Board-Data Processing

VHDL Exercises

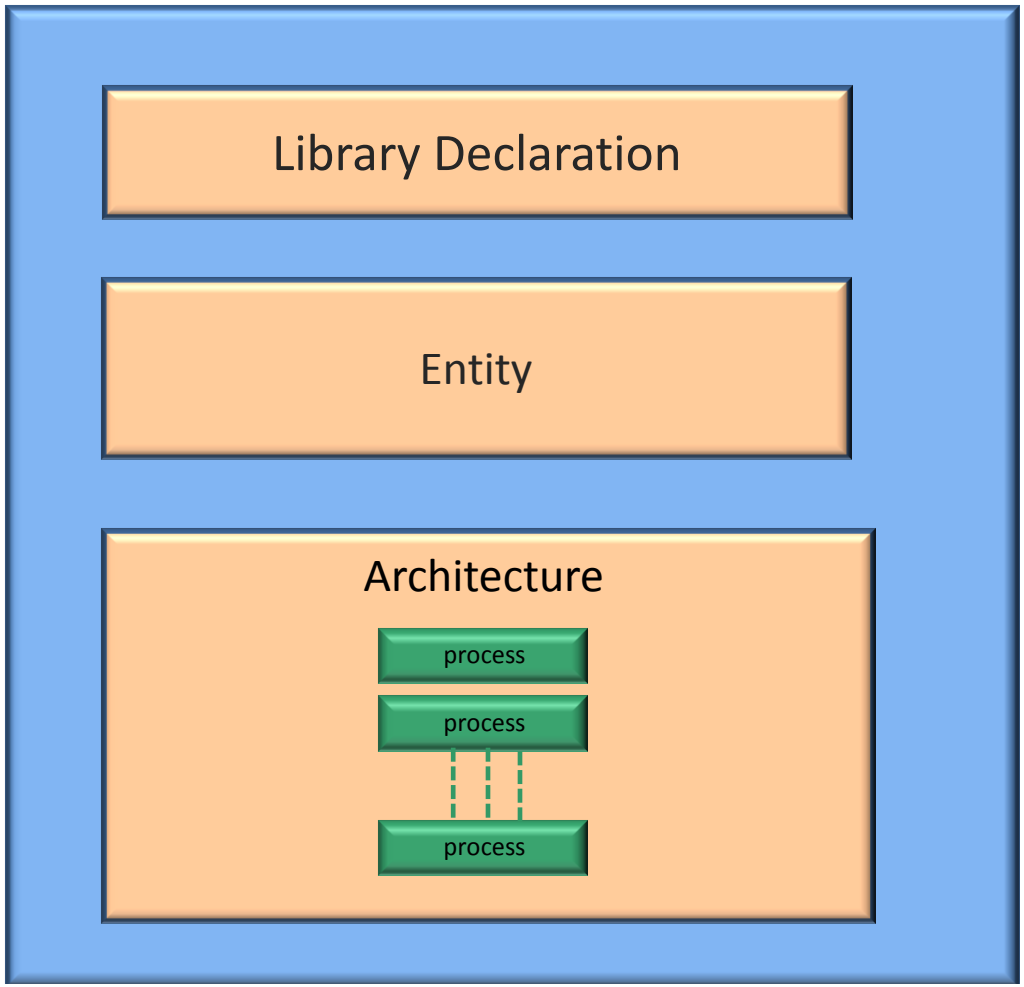
Exercise 1:

- Basics of VHDL Programming
- Stages of the Development process using FPGA's in Xilinx ISE.

Basics of VHDL

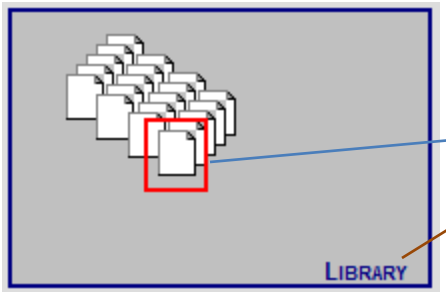
- VHDL **(Very High Speed IC Hardware description Language)** is one of the standard hardware description language used to design digital systems.

Structure of VHDL code



Structure of VHDL code

Library Declaration




```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

STANDARD: Contains all basic declarations and definitions of language constructs and it is included in all VHDL specifications by default.

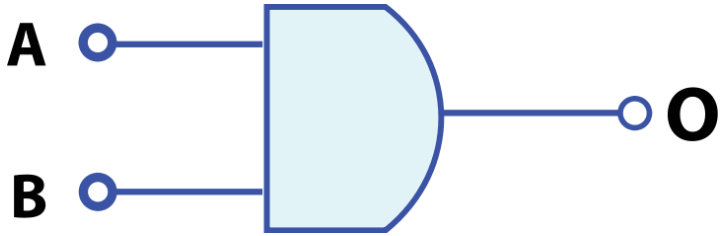
STD_LOGIC_1164: This package is not a part of the VHDL Standard but is a standard on its own; it contains the most often used language extensions.

TEXTIO: Contains declarations of basic operations on texts.

Structure of VHDL code

Entity

Defines the interface of the design

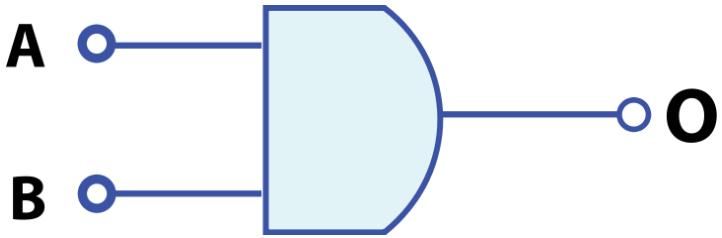


Interface

```
entity ANDGATE is
 port (
        A : in  STD_LOGIC;
        B : in  STD_LOGIC;
        O : out  STD_LOGIC);
end ANDGATE
```

Architecture

Describes the functionality of the design i.e transforming input data into output results.



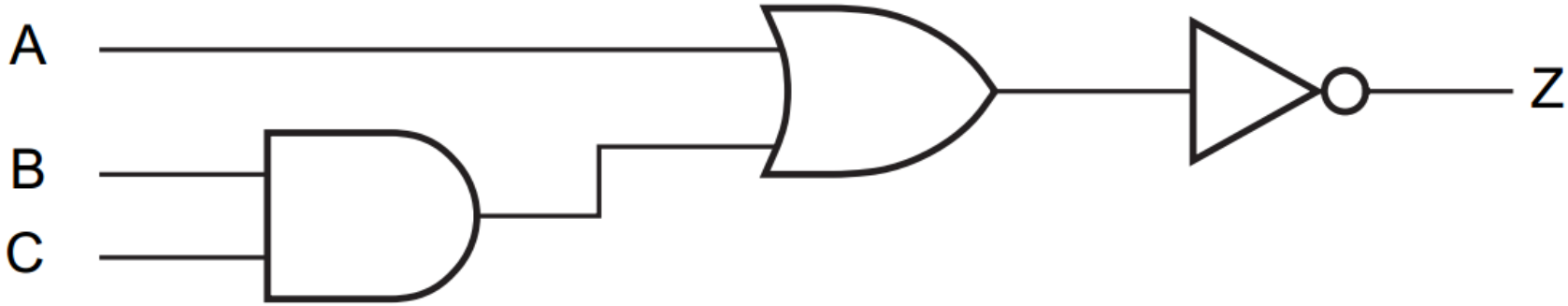
Functionality

```
architecture Behavioral of ANDGATE is
begin
    O <= A and B;
end Behavioral;
```

Example1

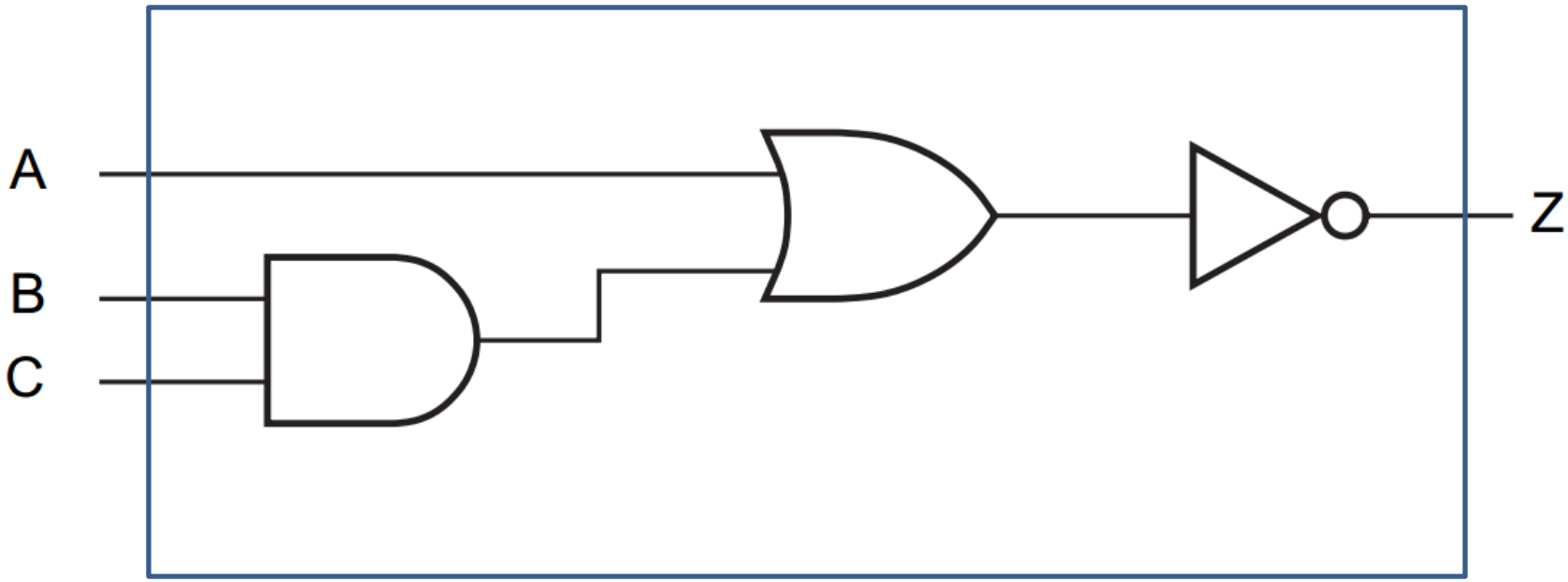
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ANDGATE is
 port (
        A : in  STD_LOGIC;
        B : in  STD_LOGIC;
        O : out  STD_LOGIC);
end ANDGATE


architecture Behavioral of ANDGATE is
begin
                O <= A and B;
end Behavioral;
```


Signals

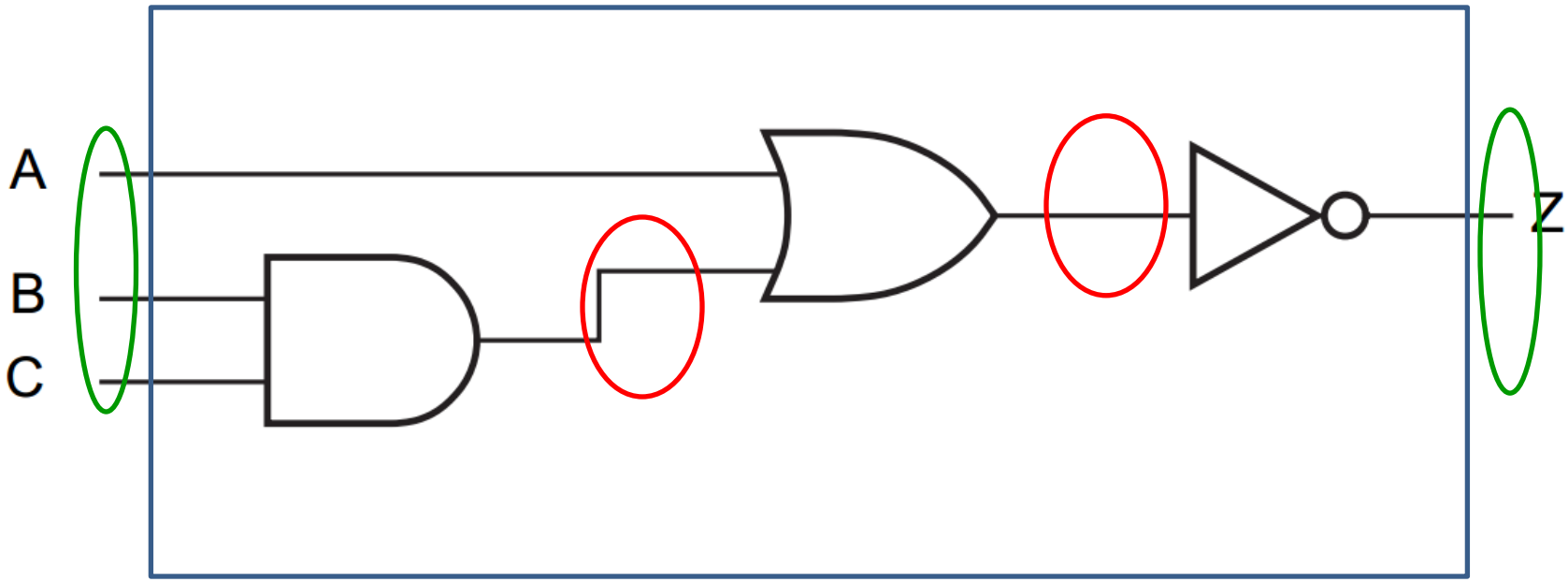


Signals

Signals



Signals

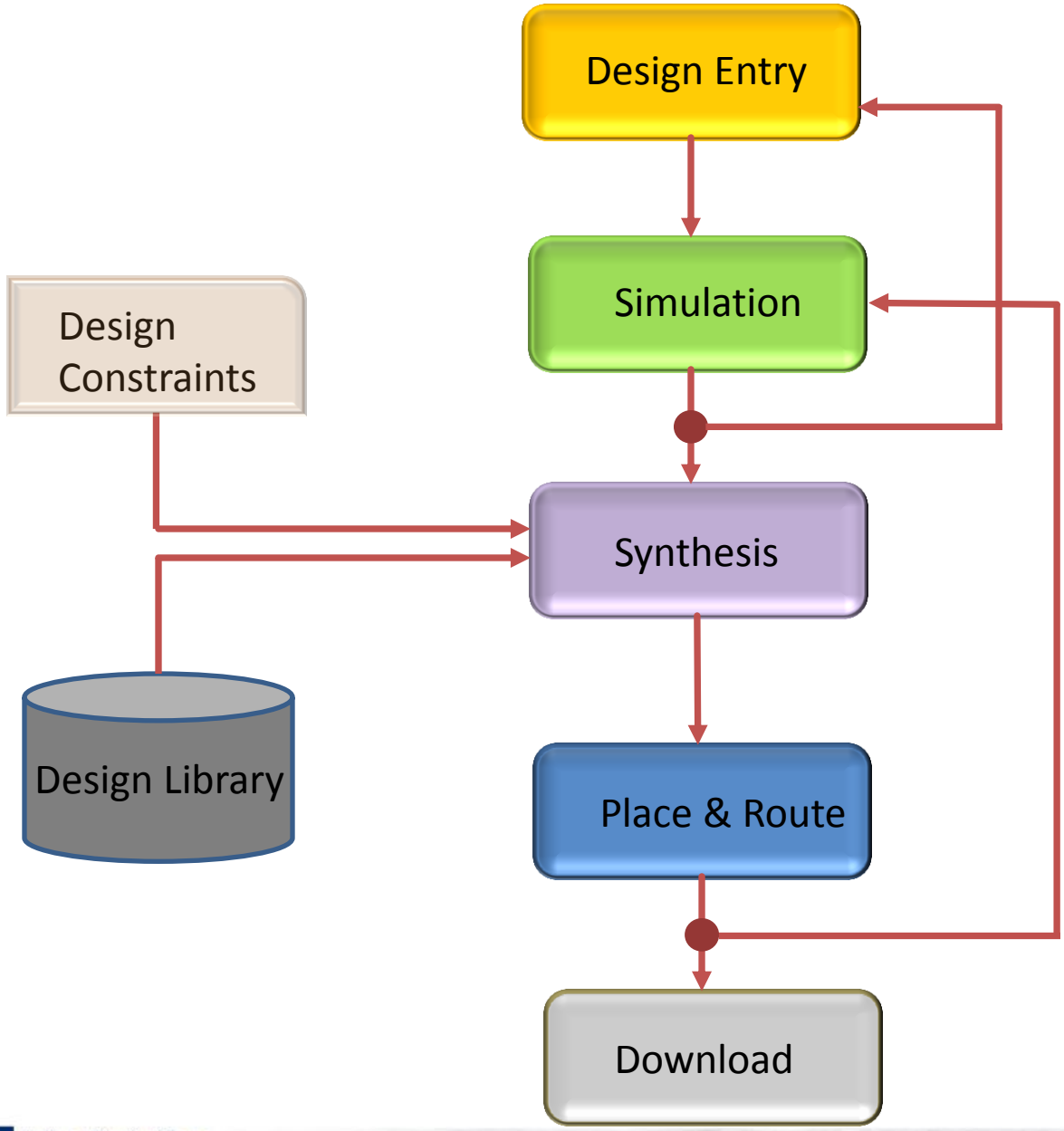


Signals

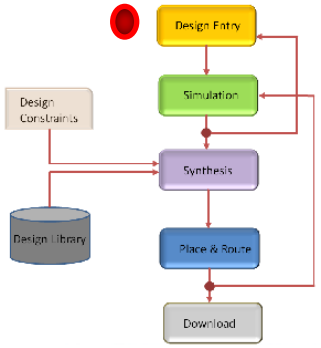
Signals represent wires or outputs of gates, FFs, etc. Ports (ins, outs, inouts) in the entity are signals. Internal signals are often needed in complex models and are declared in the architecture description as follows:

```
architecture architecture_name of entity_name is
        signal signal_name: type;
                :
                :
        signal signal_name: type;
begin
        :
        :
  end architecture architecture_name;
```


Stages of the Development of a Digital process in FPGA's


Design Entry
Simulation
Design Constraints
Synthesis
Design Library
Place & Route
Download

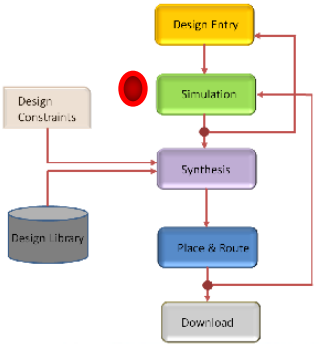
Design Entry



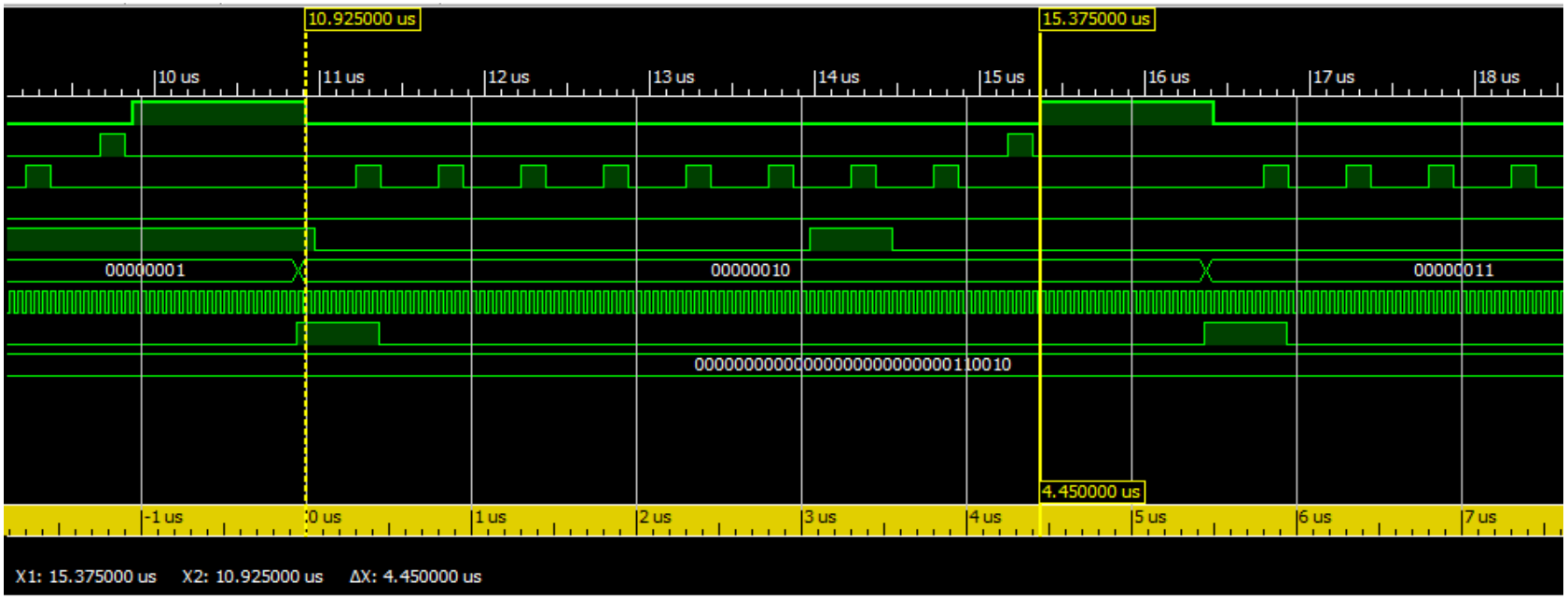
-> Writing the VHDL or VERILOG code

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity timer is
7      Port (  clk : in  STD_LOGIC;
8           reset : in STD_LOGIC;
9           interrupt: in STD_LOGIC;
10          idle : inout STD_LOGIC;
11          sec : inout STD_LOGIC_VECTOR(4 downto 0));
12 end timer;
13 architecture Behavioral of timer is
14    signal cnt_int : std_logic_vector(4 downto 0);
15 begin
16       process (clk)
17       begin
18          if clk='1' and clk'event then
19             if (reset='1') then
20                idle <= '0';
21                cnt_int <= "00000";
22                sec <= "00000";
23             end if;
24             if (interrupt = '0' and not(sec = "10000")) then
25                cnt_int <= cnt_int + 1;
26                if cnt_int="10010" then
27                   sec <= sec + 1;
28                   cnt_int <= "00000";
29                end if;
30             end if;
31             if sec="10000" then
32                   idle <= '1';
33             else
34                   idle <= '0';
35             end if;
36          end if;
37       end process;
38 end Behavioral;
```

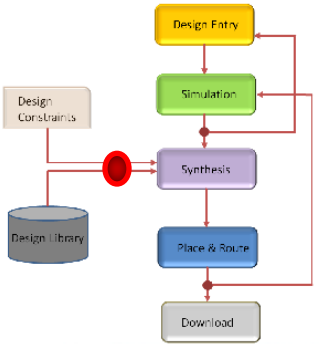
Simulation



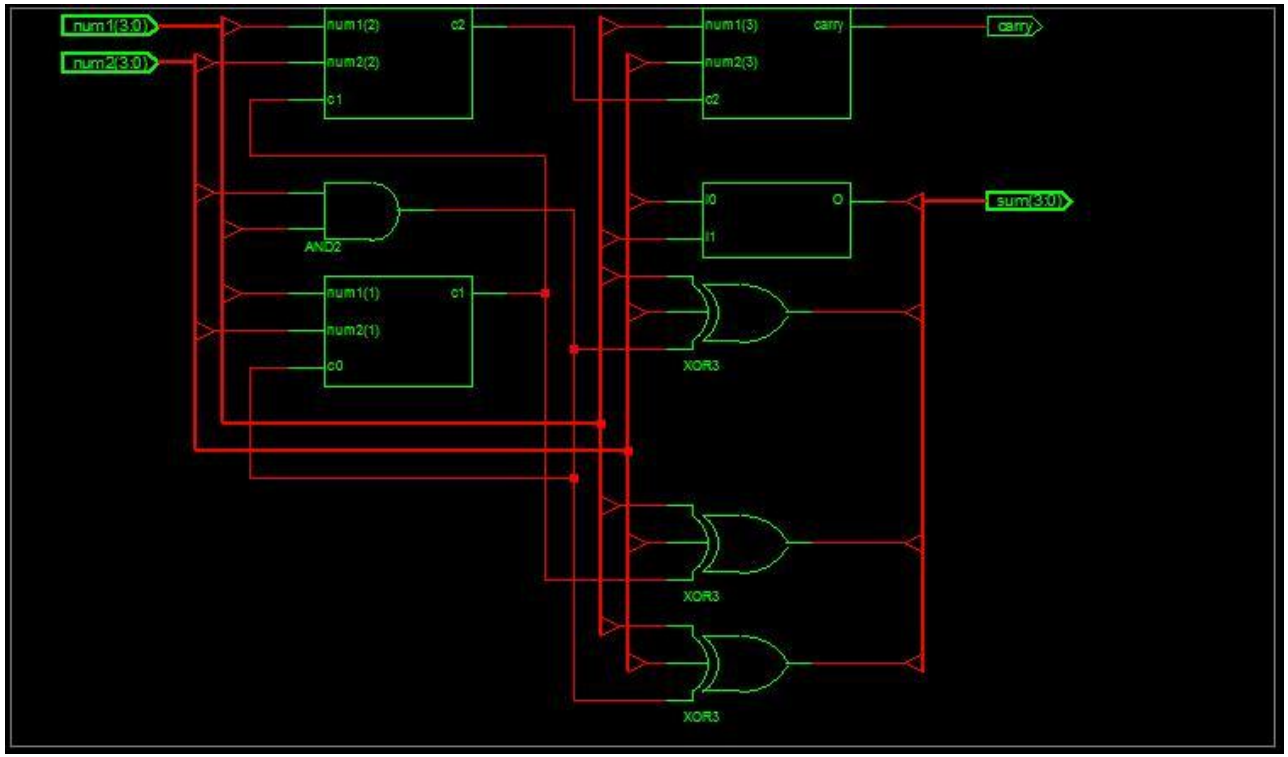
-> Identify Syntax and logic errors



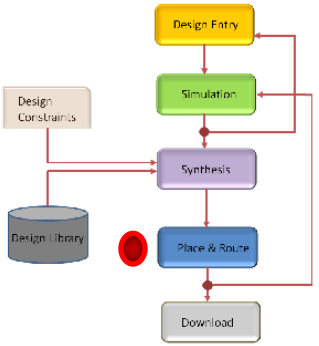
Synthesis



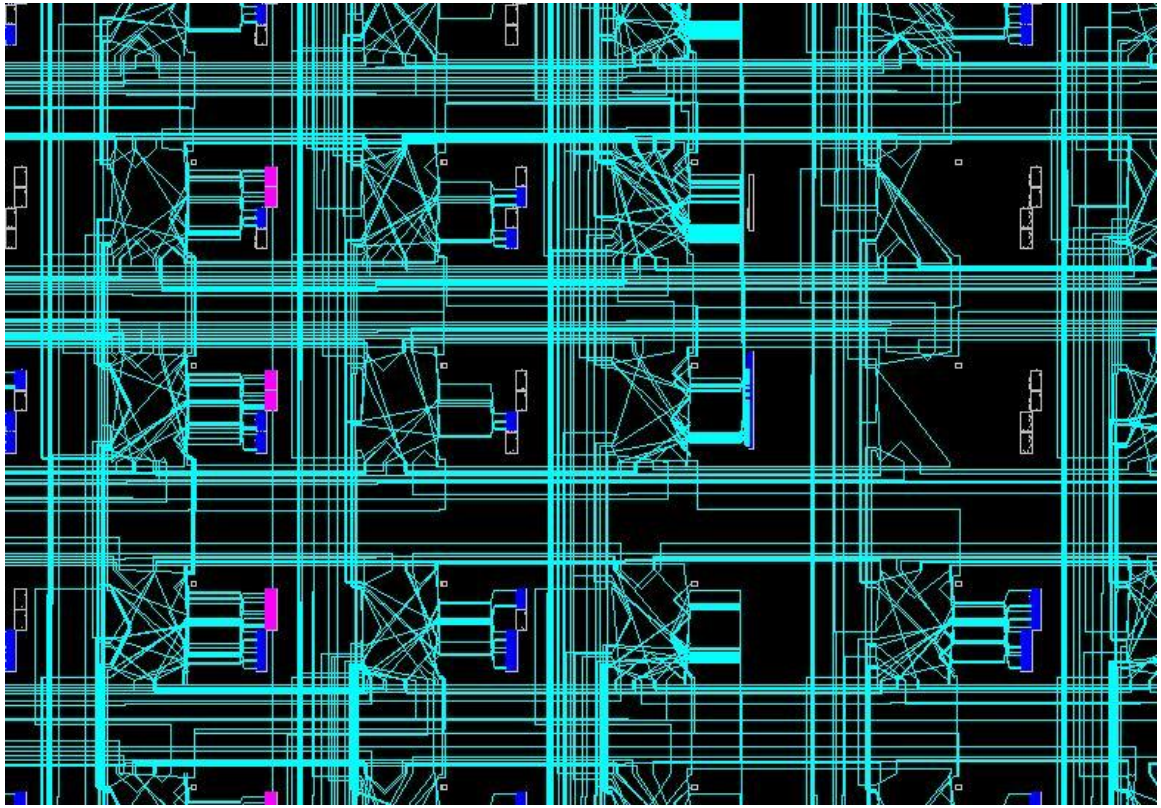
-> Generate Netlist



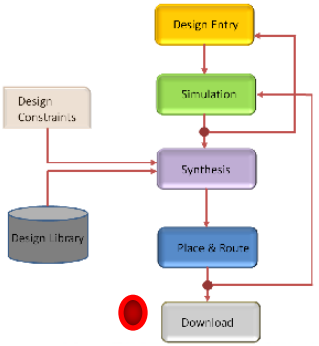
Place & Route



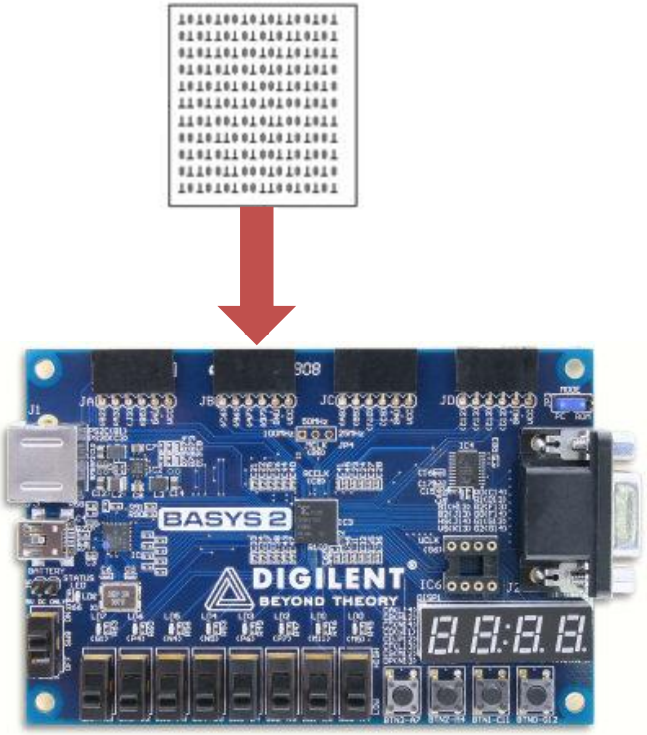
-> Map, Place & Route and generate the binary file



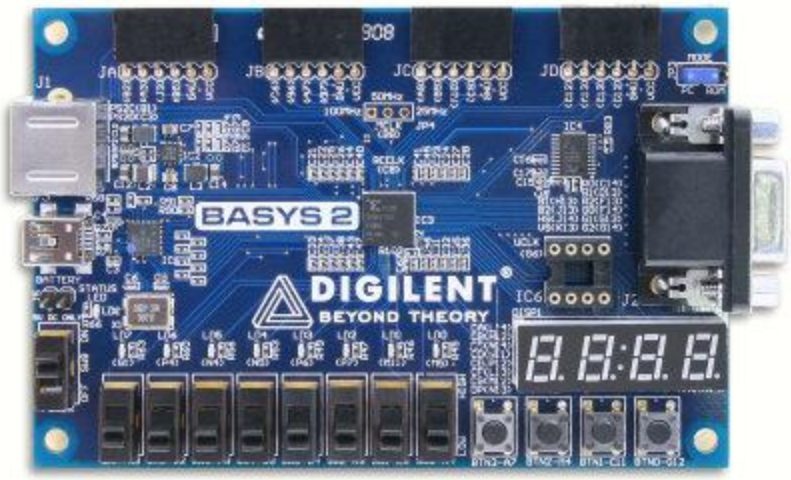
Download



-> Flash the binary file on the FPGA development board



FPGAs Kit and Development Enviroment

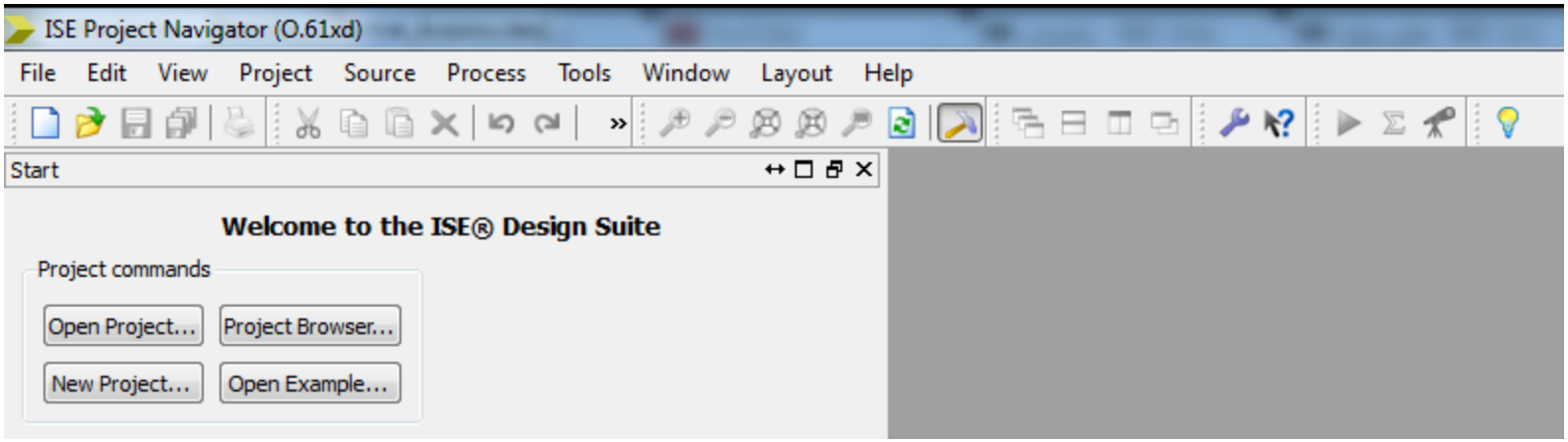


XILINX®

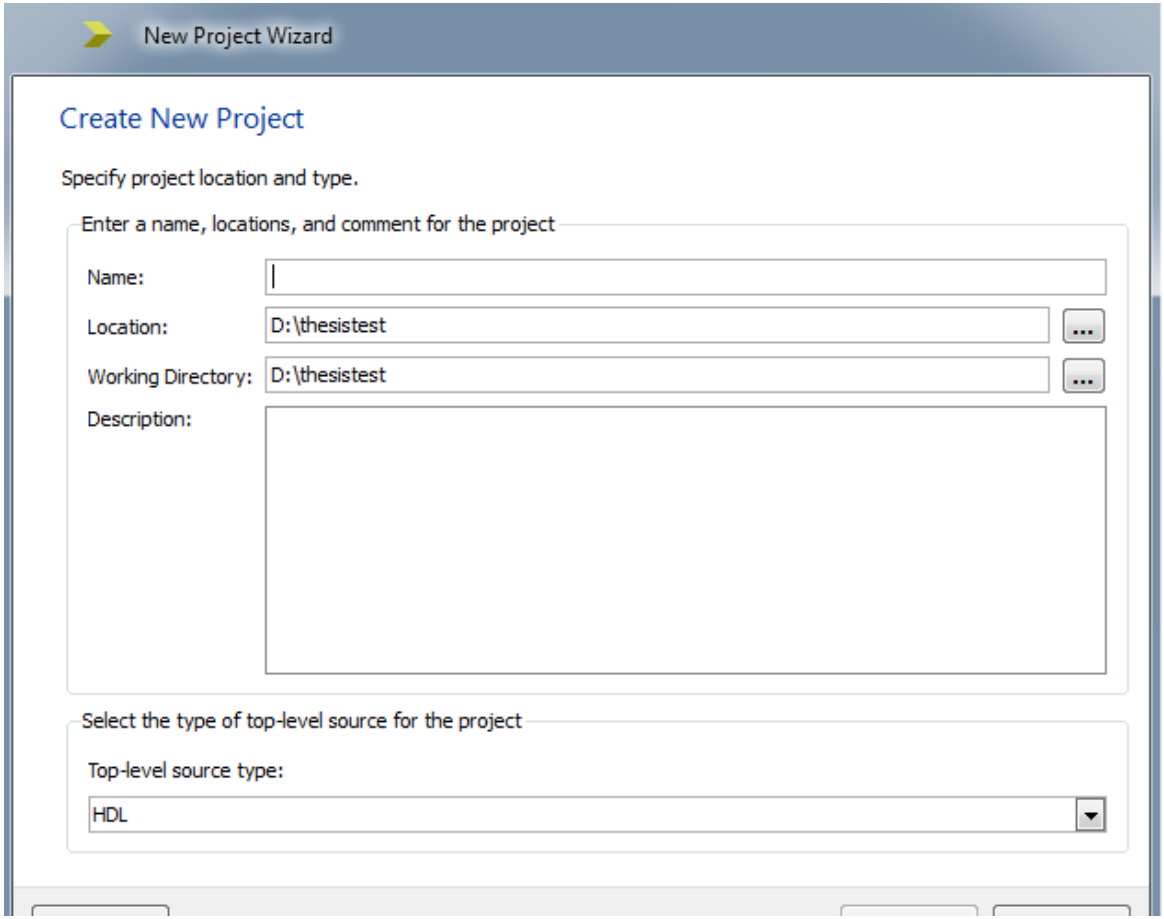
XILINX®
ISE
DESIGN SUITE

How to Start the Hardware Development in FPGA

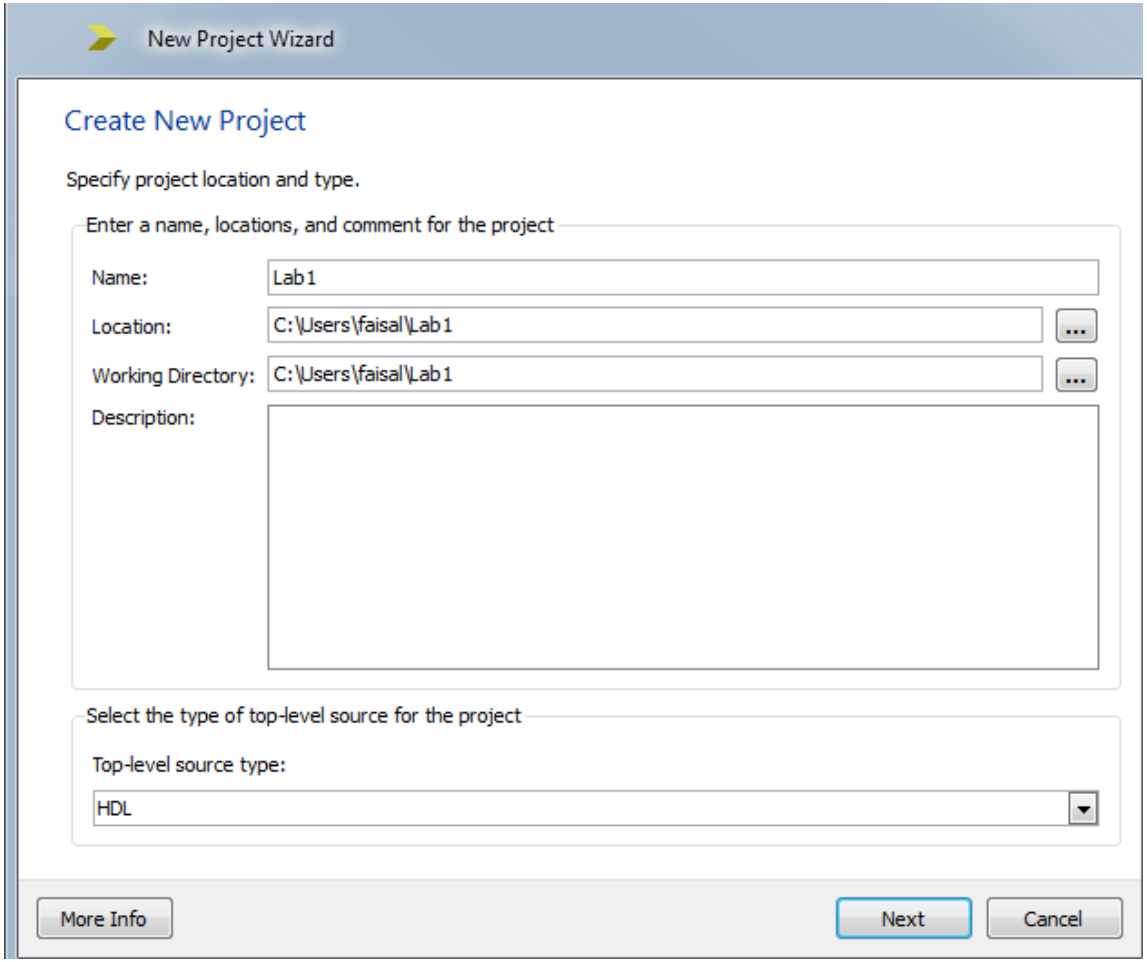
1.Open the Xilinx ISE project Navigator



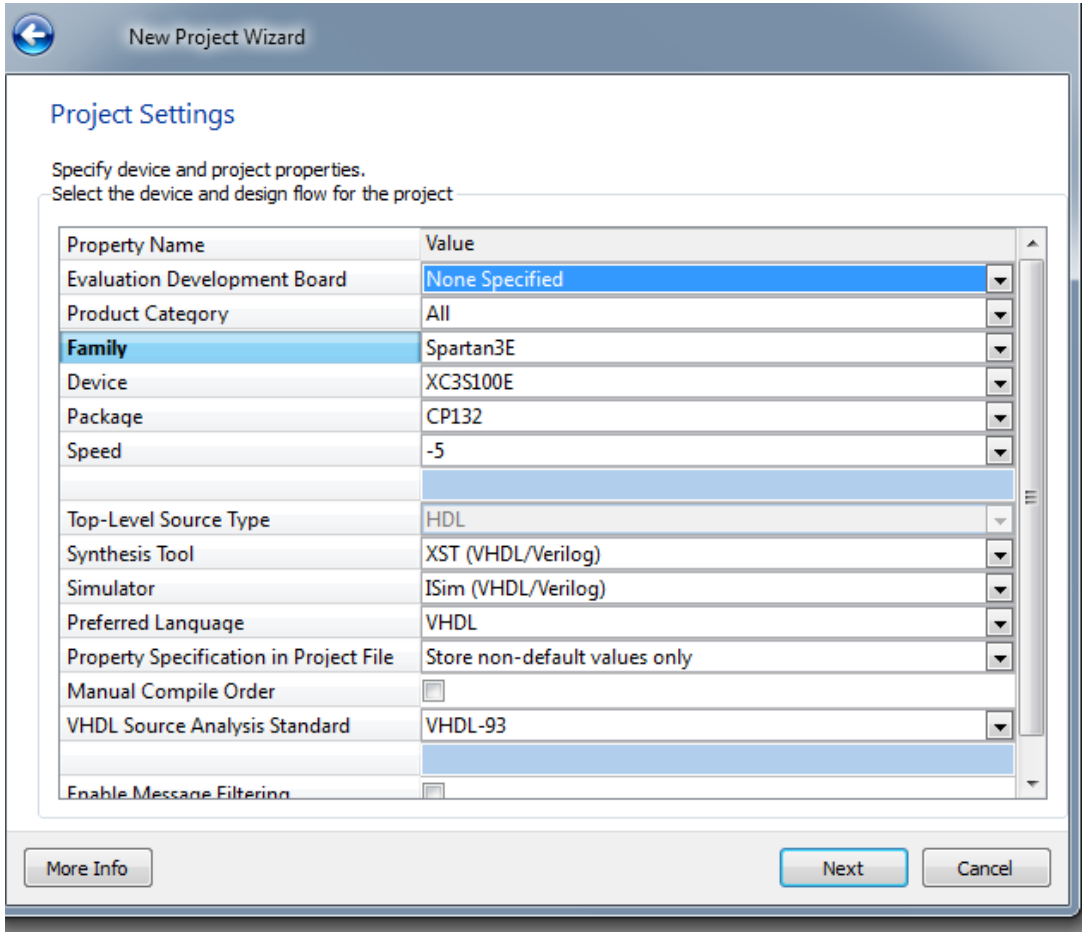
2. Create a New Project and save it to appropriate location and please note that don’t use spaces in the destination of the project



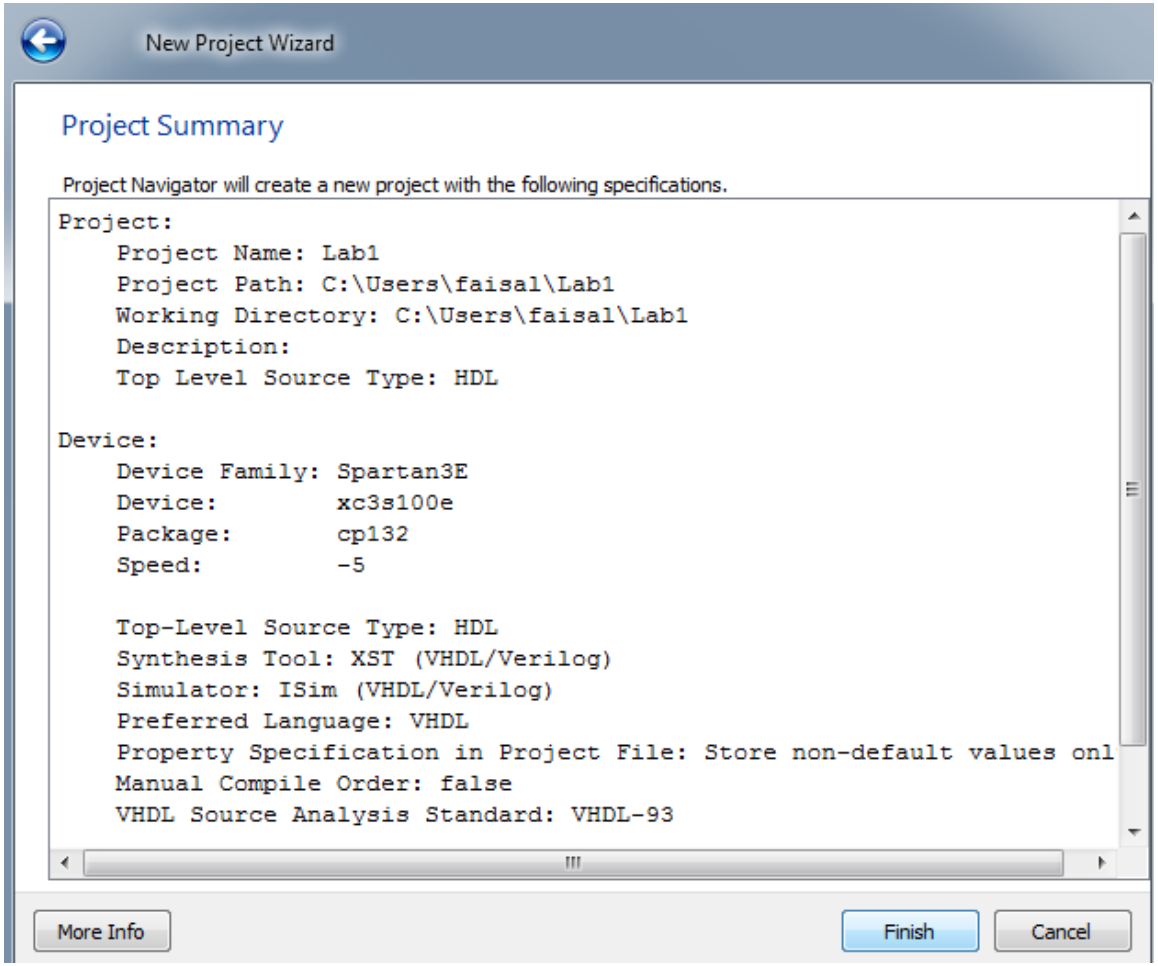
3. Give the name of the project and click Next.



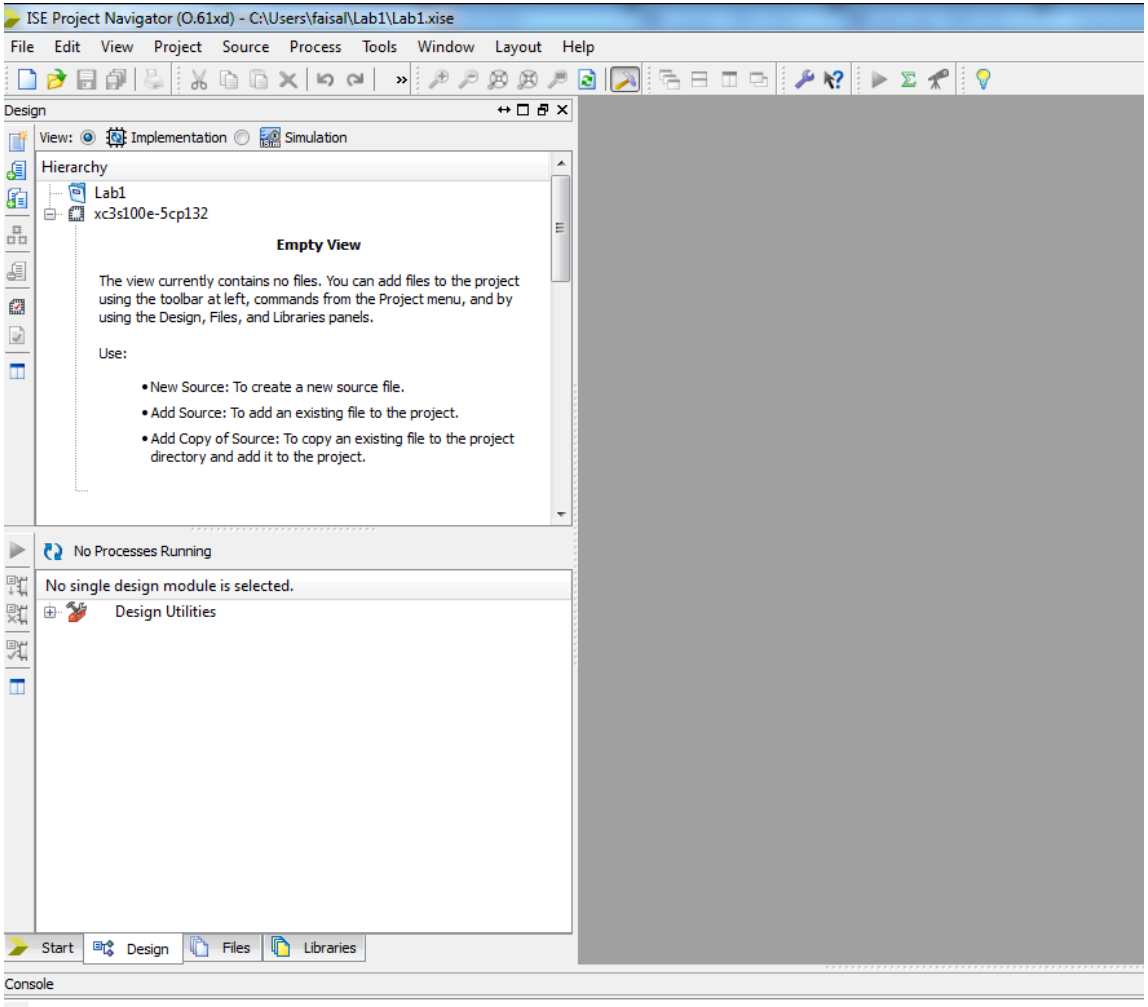
4. Select the Family, Device, Package, Simulator and Preferred Language as shown below.



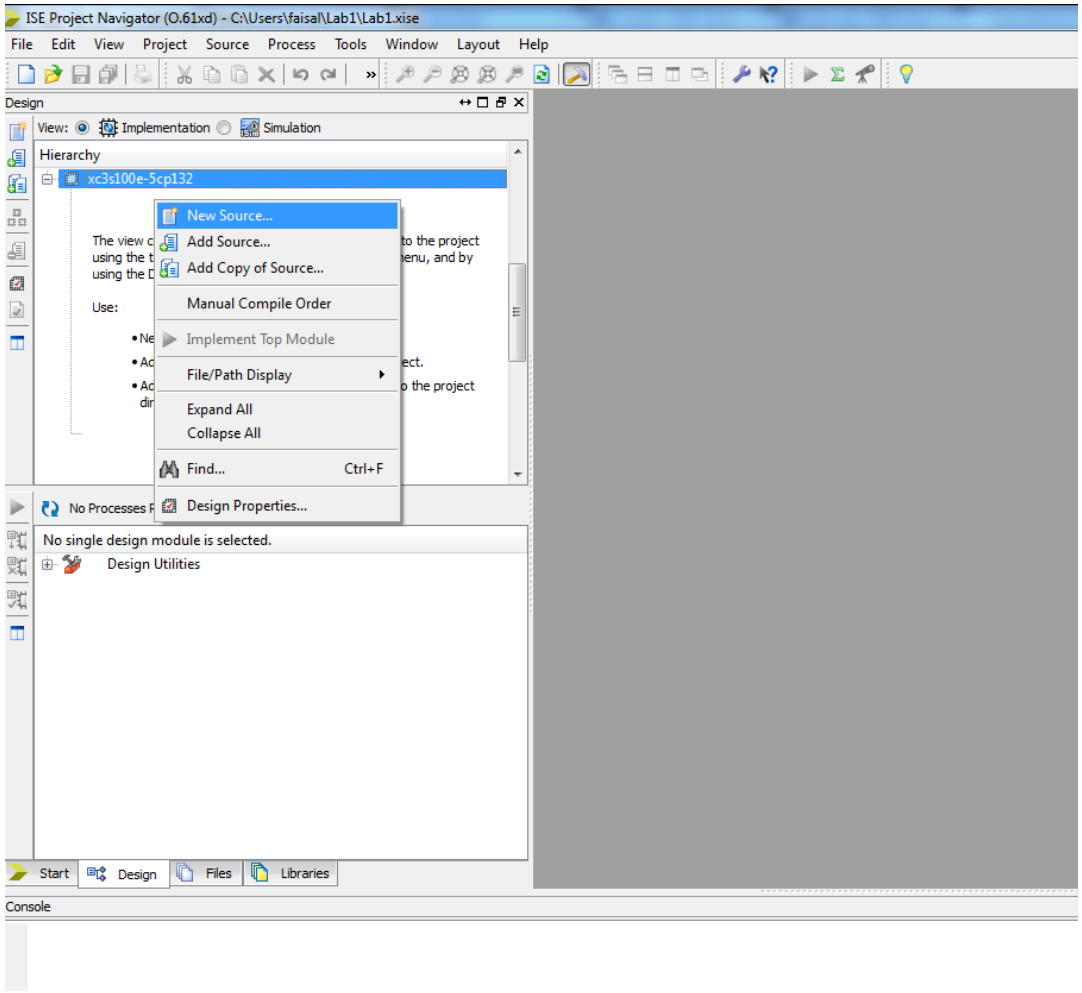
5. Click Finish.



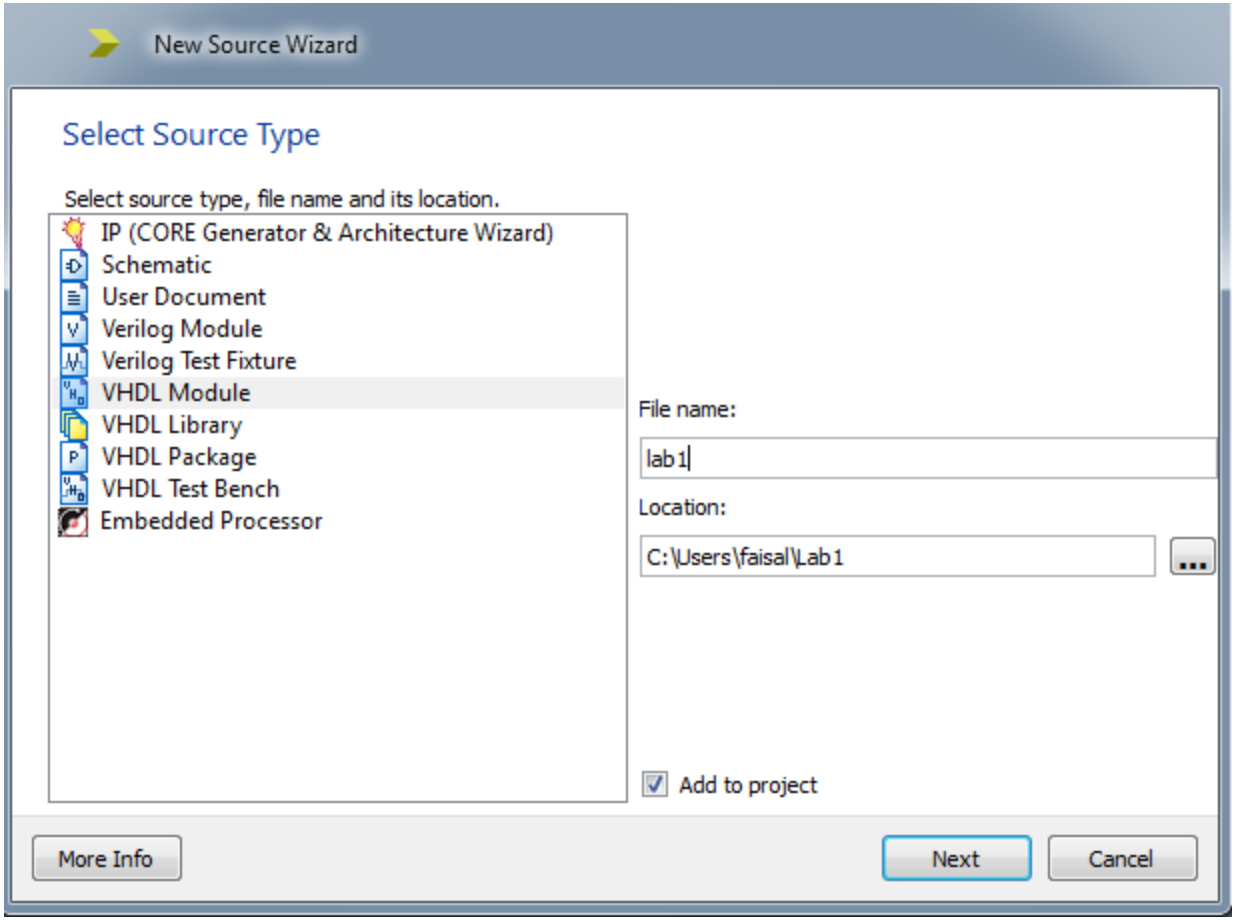
6. Now you have to Add a new vhdl file to write the code



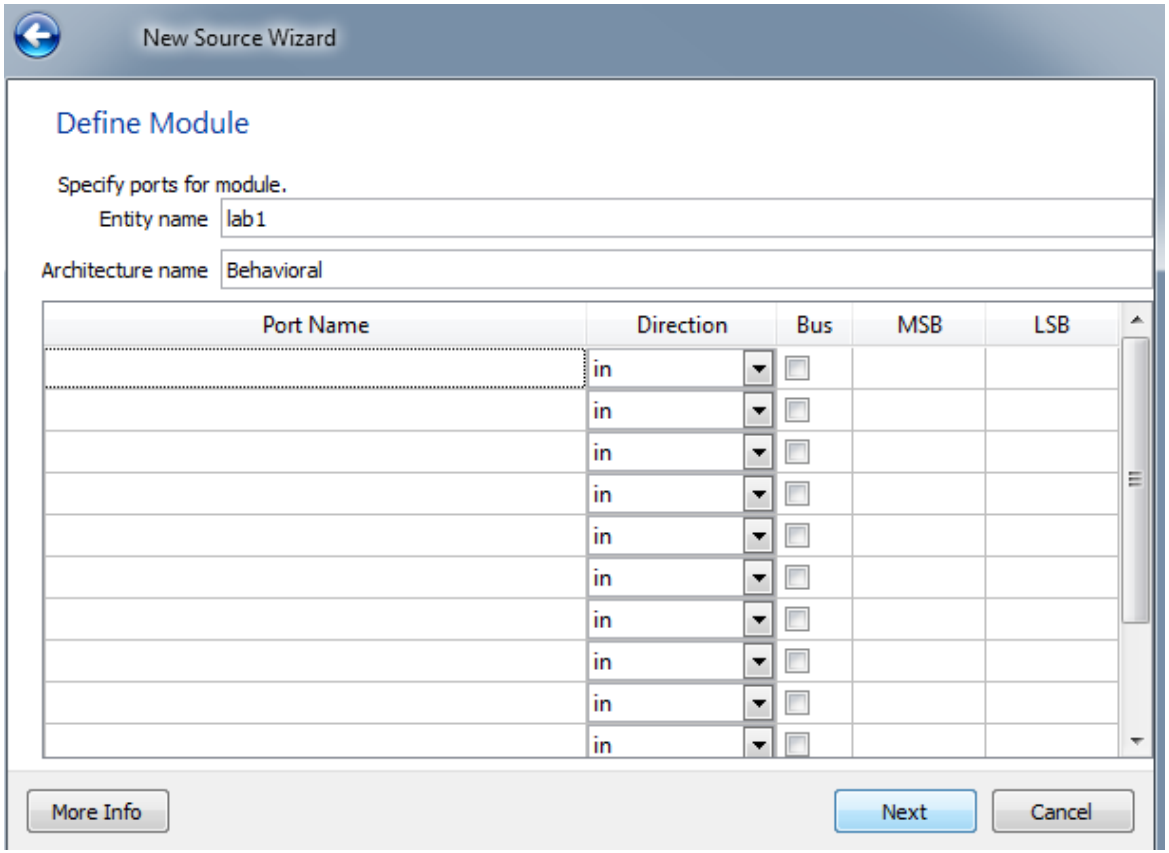
7.Right Click the on the name of the device.

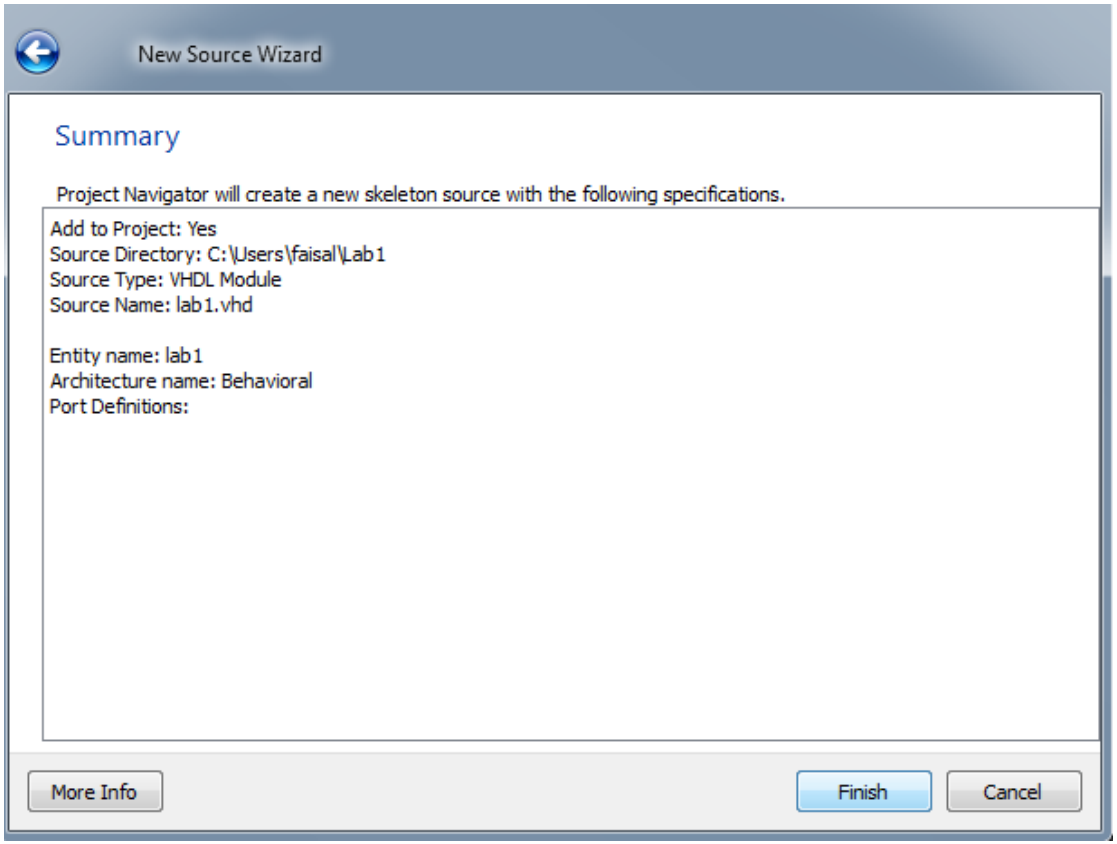


4. Select the VHDL module as Source type and name the VHDL file

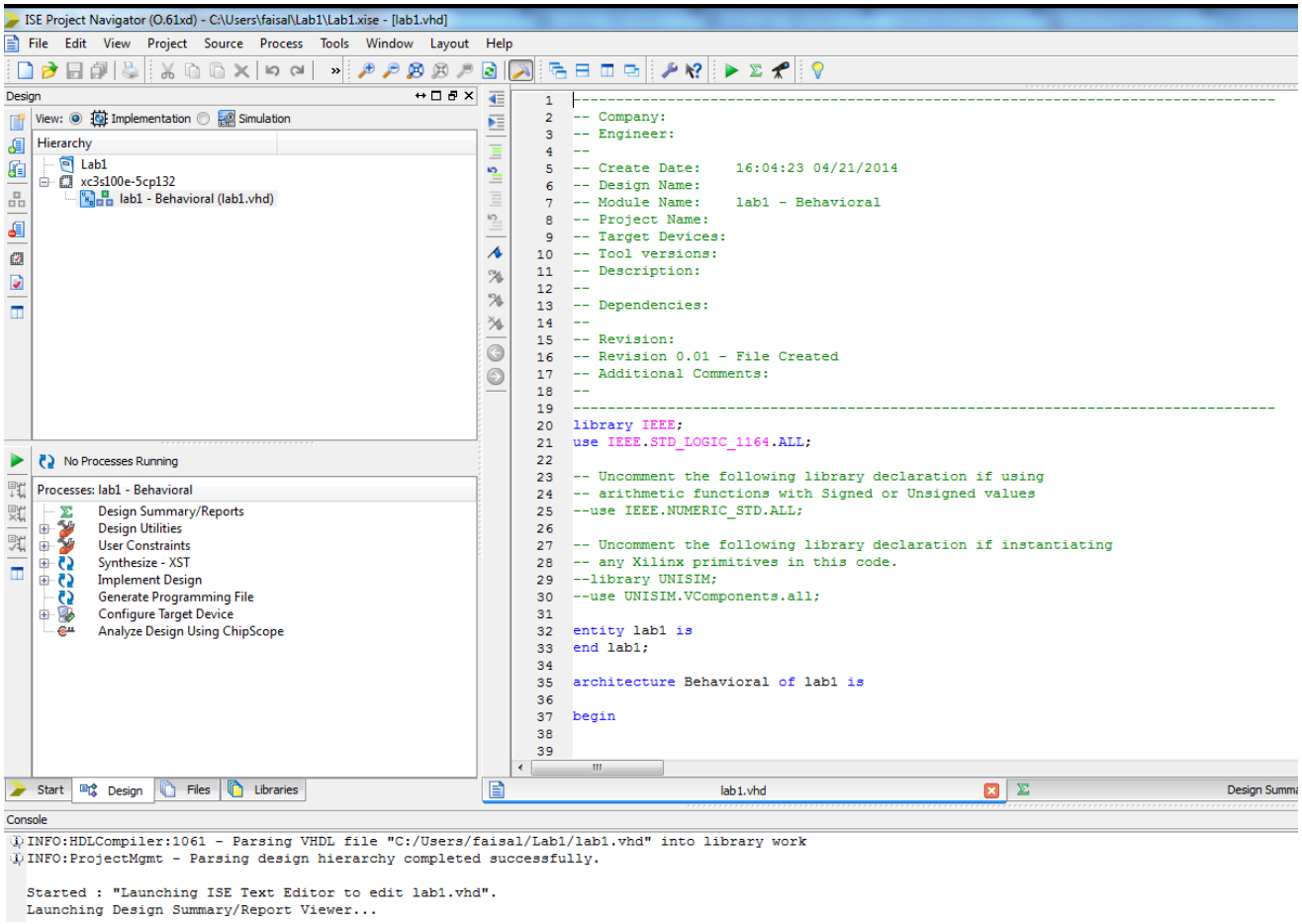


8. Click Next

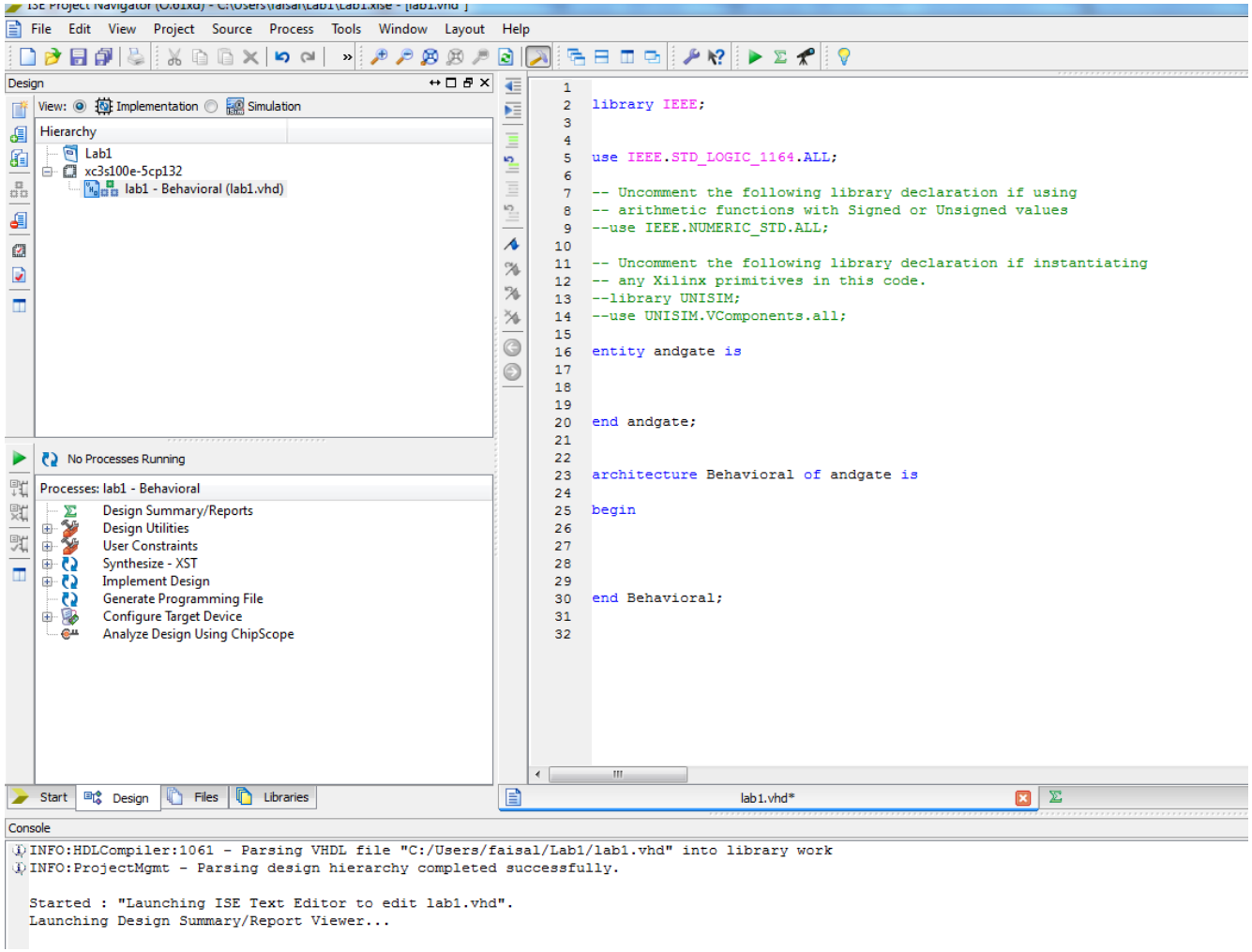

New Source Wizard
Define Module
Specify ports for module.
Entity name
lab1
Architecture name
Behavioral
Port Name
Direction
Bus
MSB
LSB
in
More Info
Next
Cancel


New Source Wizard
Summary
Project Navigator will create a new skeleton source with the following specifications.
Add to Project: Yes
Source Directory: C:\Users\faisal\Lab1
Source Type: VHDL Module
Source Name: lab1.vhd
Entity name: lab1
Architecture name: Behavioral
Port Definitions:
More Info
Finish
Cancel

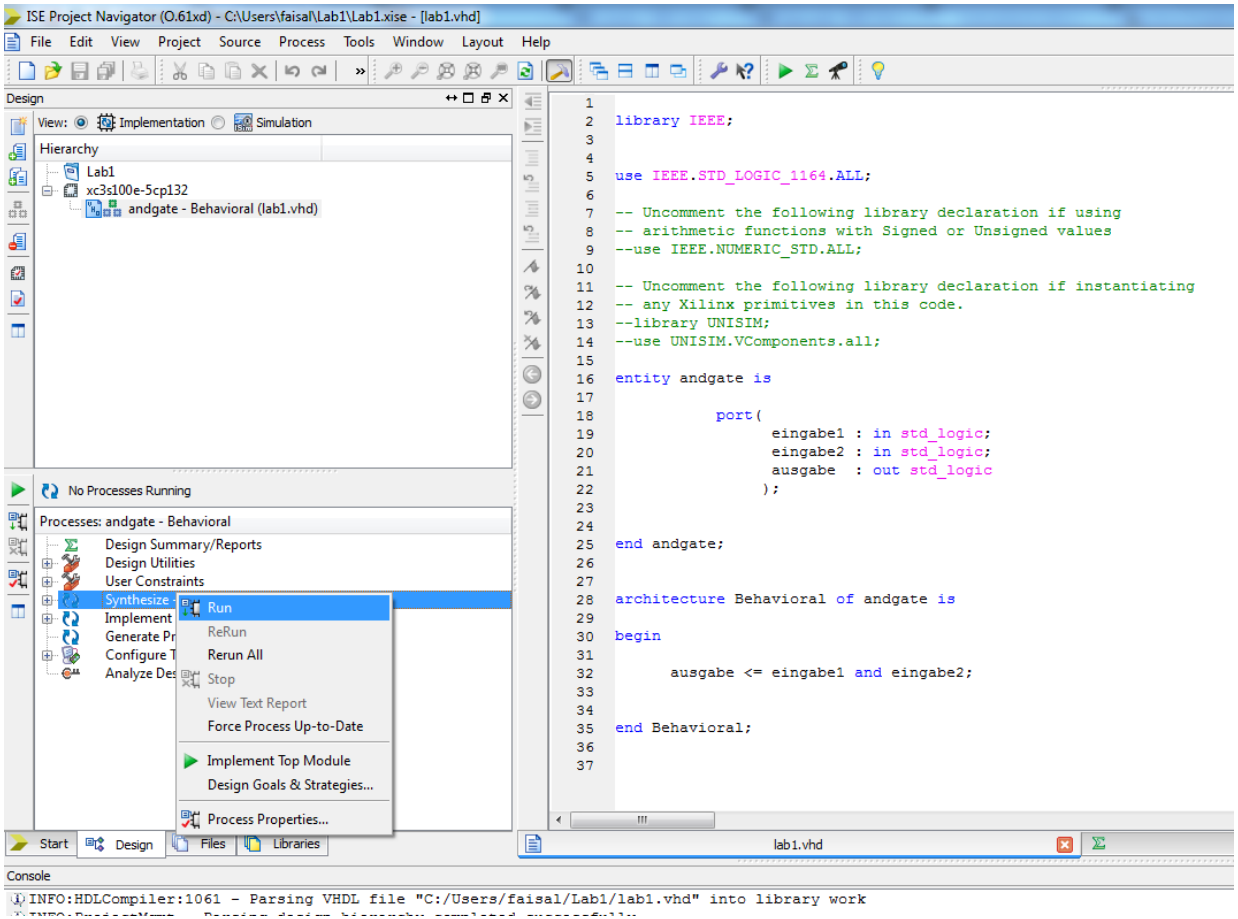
9. Open the VHDL file in the editor.



10. You can write your VHDL code here .

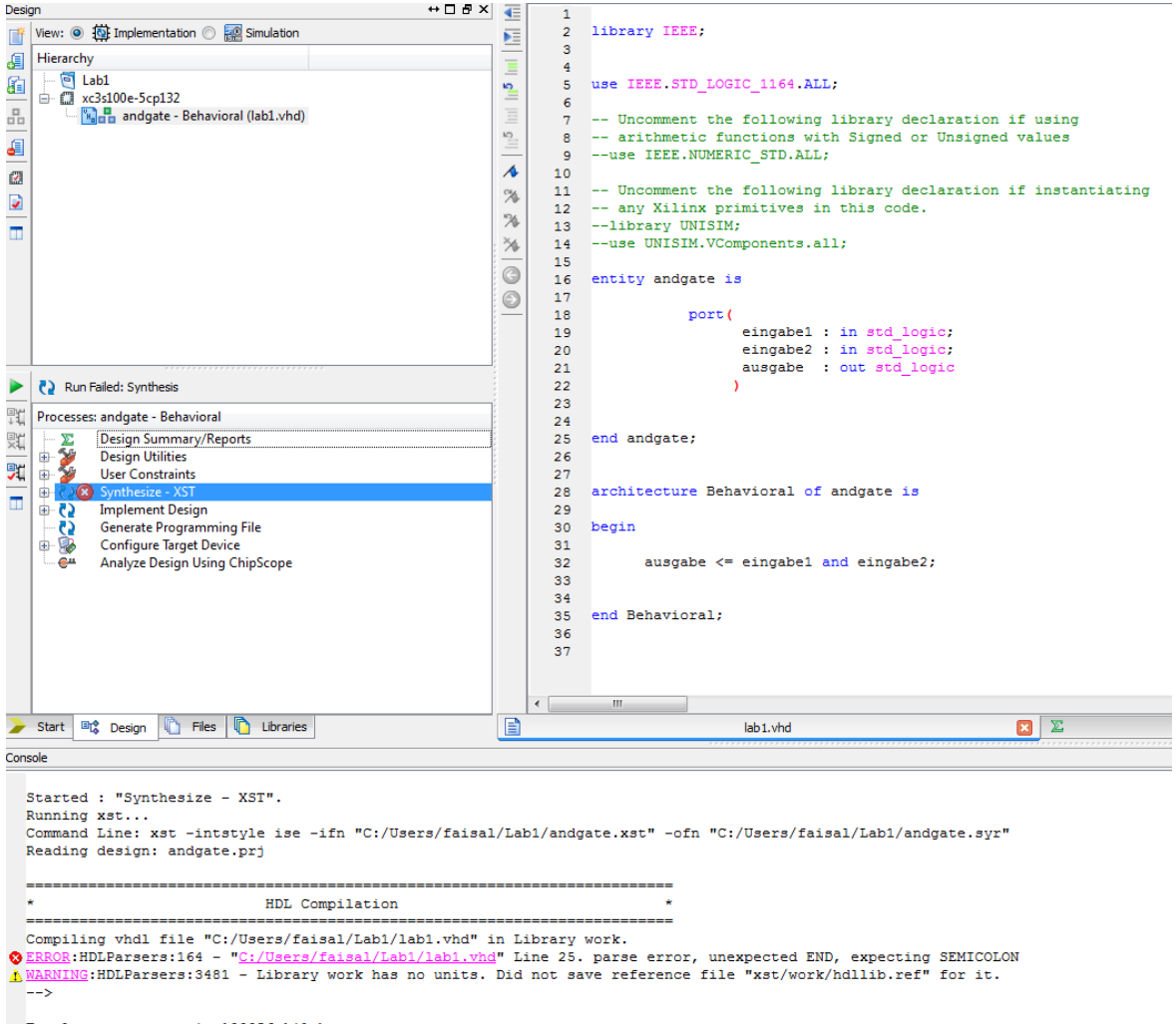


11. Write click on Synthesis option and select run.



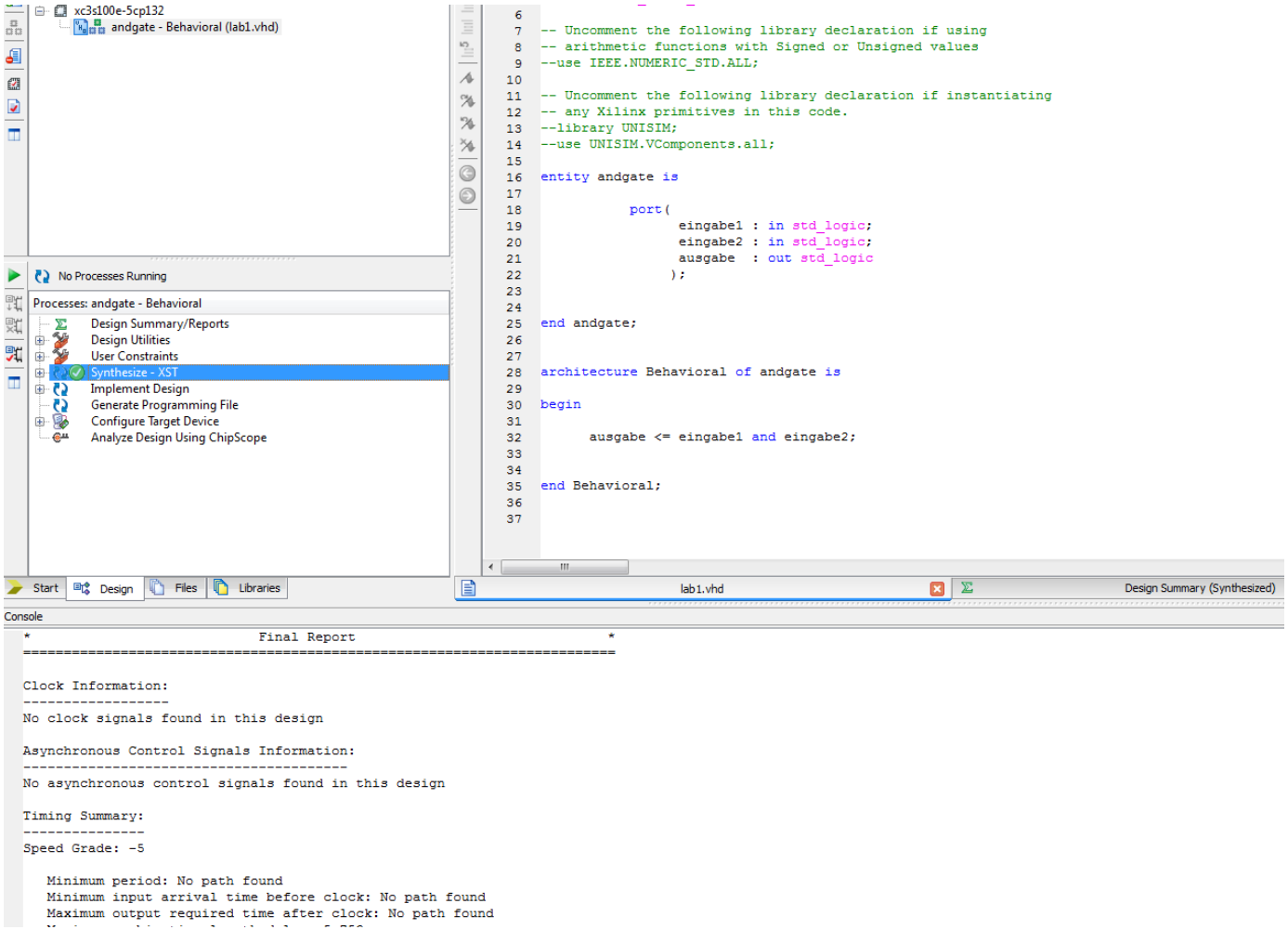

```
1
2  library IEEE;
3
4
5  use IEEE.STD_LOGIC_1164.ALL;
6
7  -- Uncomment the following library declaration if using
8  -- arithmetic functions with Signed or Unsigned values
9  --use IEEE.NUMERIC_STD.ALL;
10
11 -- Uncomment the following library declaration if instantiating
12 -- any Xilinx primitives in this code.
13 --library UNISIM;
14 --use UNISIM.VComponents.all;
15
16 entity andgate is
17
18           port(
19                 eingabe1 : in std_logic;
20                 eingabe2 : in std_logic;
21                 ausgabe  : out std_logic
22               );
23
24
25 end andgate;
26
27
28 architecture Behavioral of andgate is
29
30 begin
31
32       ausgabe <= eingabe1 and eingabe2;
33
34
35 end Behavioral;
36
37
```

12.If you have some error in your VHDL code it will be Red otherwise green.



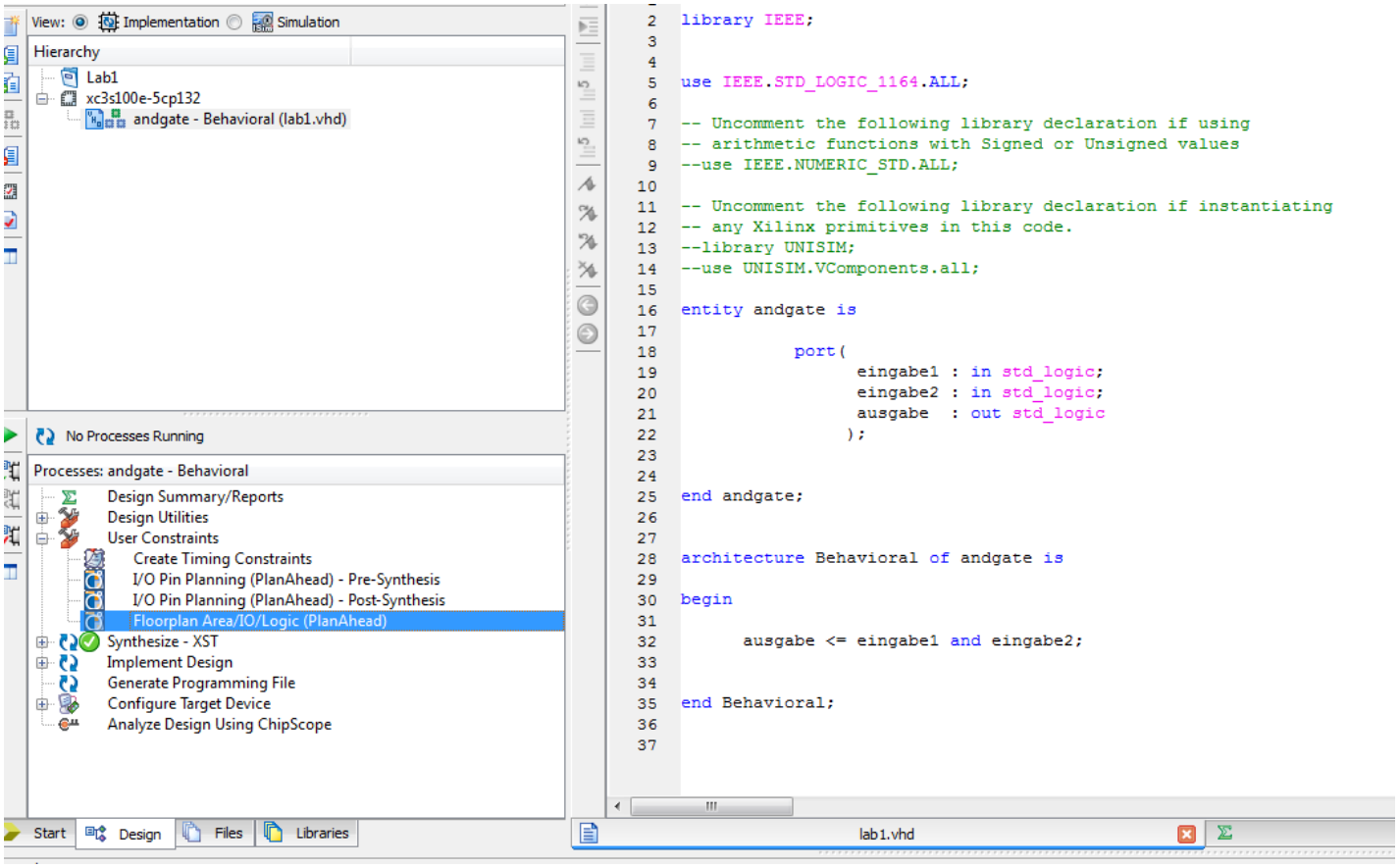

```
1
2   library IEEE;
3
4
5   use IEEE.STD_LOGIC_1164.ALL;
6
7   -- Uncomment the following library declaration if using
8   -- arithmetic functions with Signed or Unsigned values
9   --use IEEE.NUMERIC_STD.ALL;
10
11  -- Uncomment the following library declaration if instantiating
12  -- any Xilinx primitives in this code.
13  --library UNISIM;
14  --use UNISIM.VComponents.all;
15
16  entity andgate is
17
18            port(
19                  eingabe1 : in std_logic;
20                  eingabe2 : in std_logic;
21                  ausgabe  : out std_logic
22                 )
23
24
25  end andgate;
26
27
28  architecture Behavioral of andgate is
29
30  begin
31
32        ausgabe <= eingabe1 and eingabe2;
33
34
35  end Behavioral;
36
37
```

13. Synthesis Process is Finished.

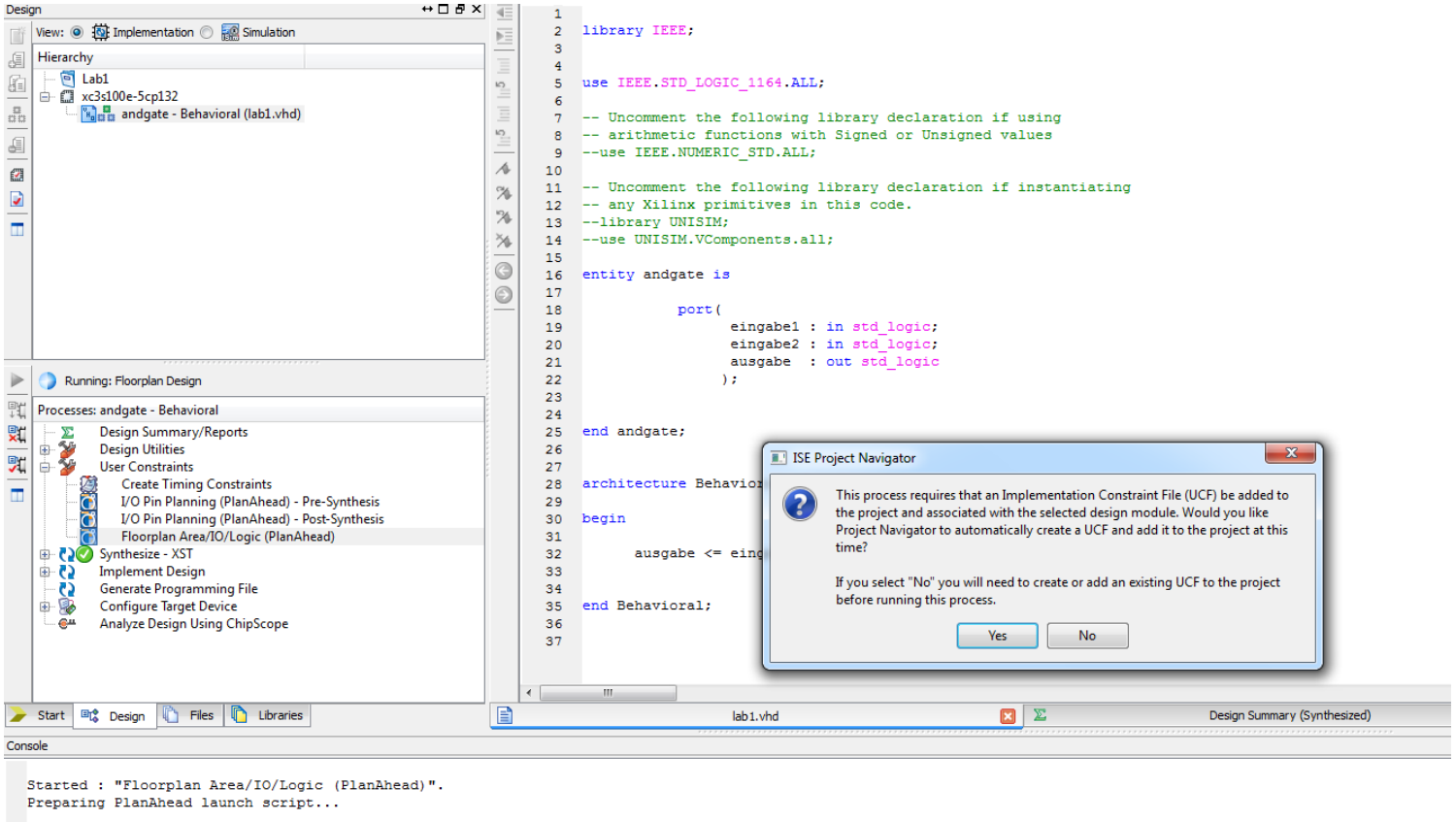



```
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity andgate is

          port(
                eingabe1 : in std_logic;
                eingabe2 : in std_logic;
                ausgabe  : out std_logic
               );


end andgate;


architecture Behavioral of andgate is

begin

     ausgabe <= eingabe1 and eingabe2;


end Behavioral;
```

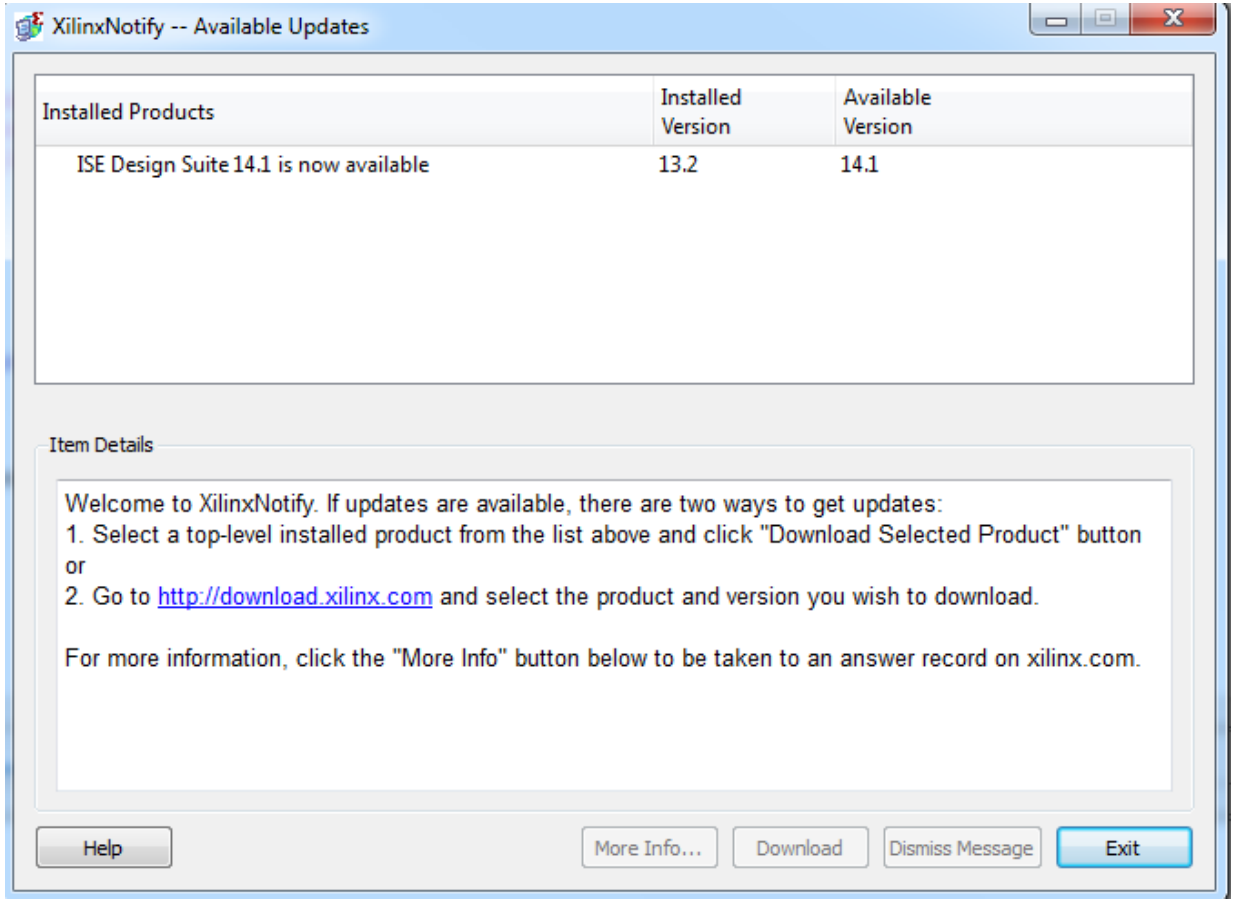
14. Now Pin Assignment has to be done so Click on FloorPlanning



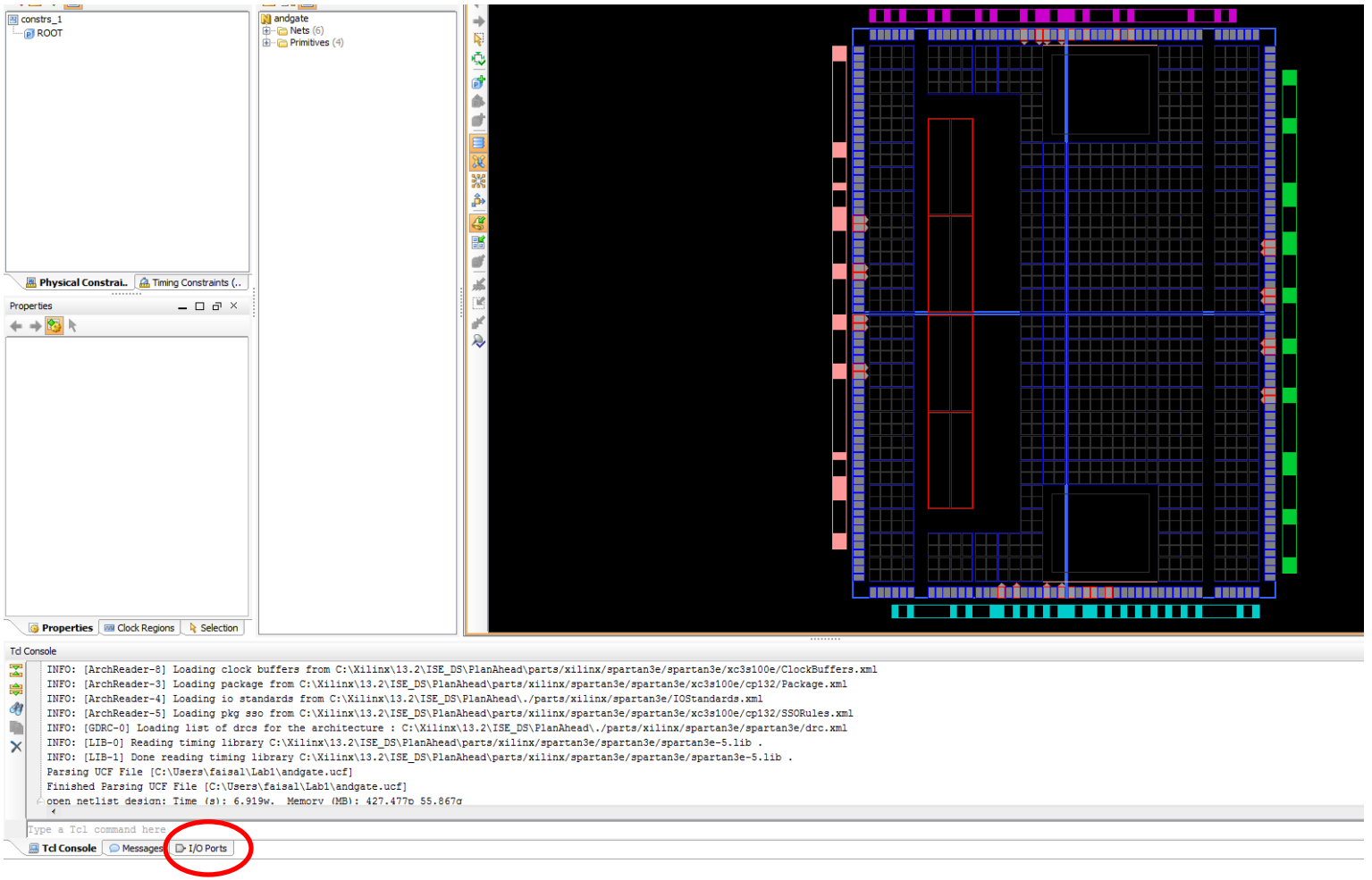
15. Click Yes and Wait , It takes some time to open the window for floor planning (Pin Assignment)



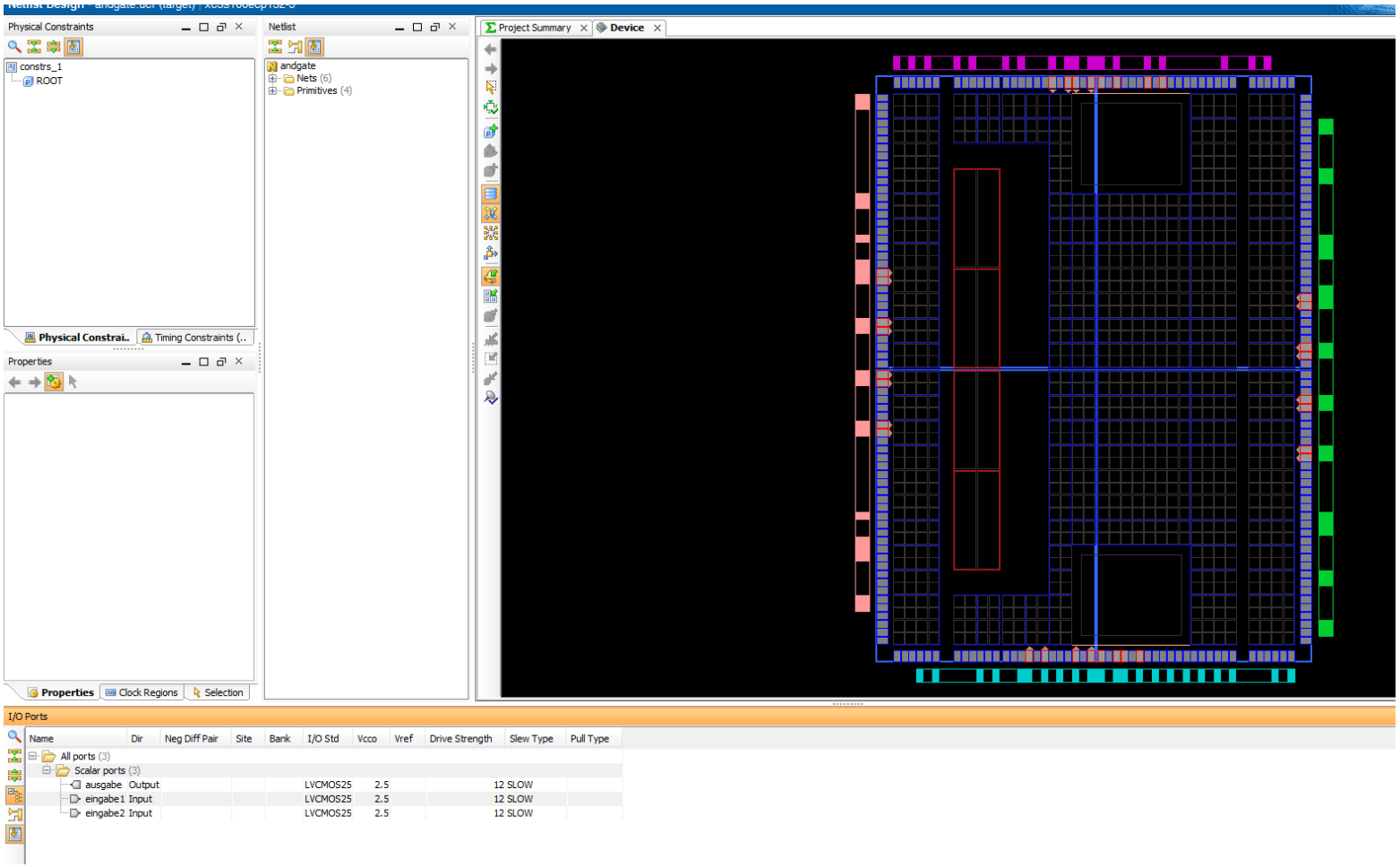
16.Don’t update the software so press Exit button to cancel the update.



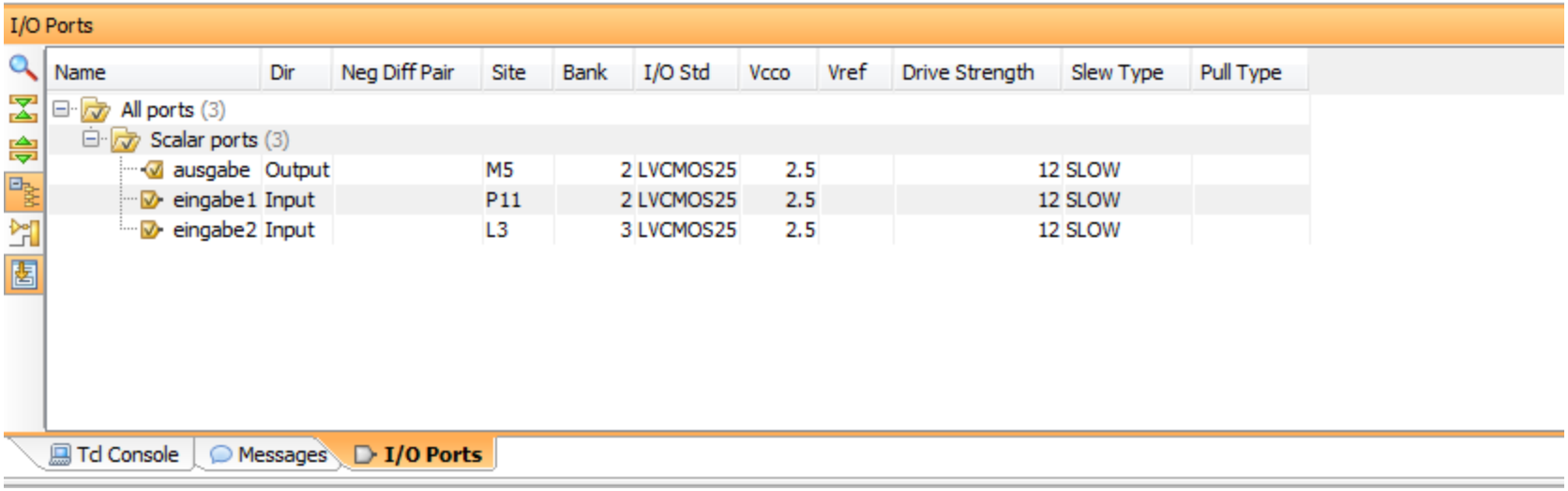
17. This is place where you will map your I/O’s to the actual hardware



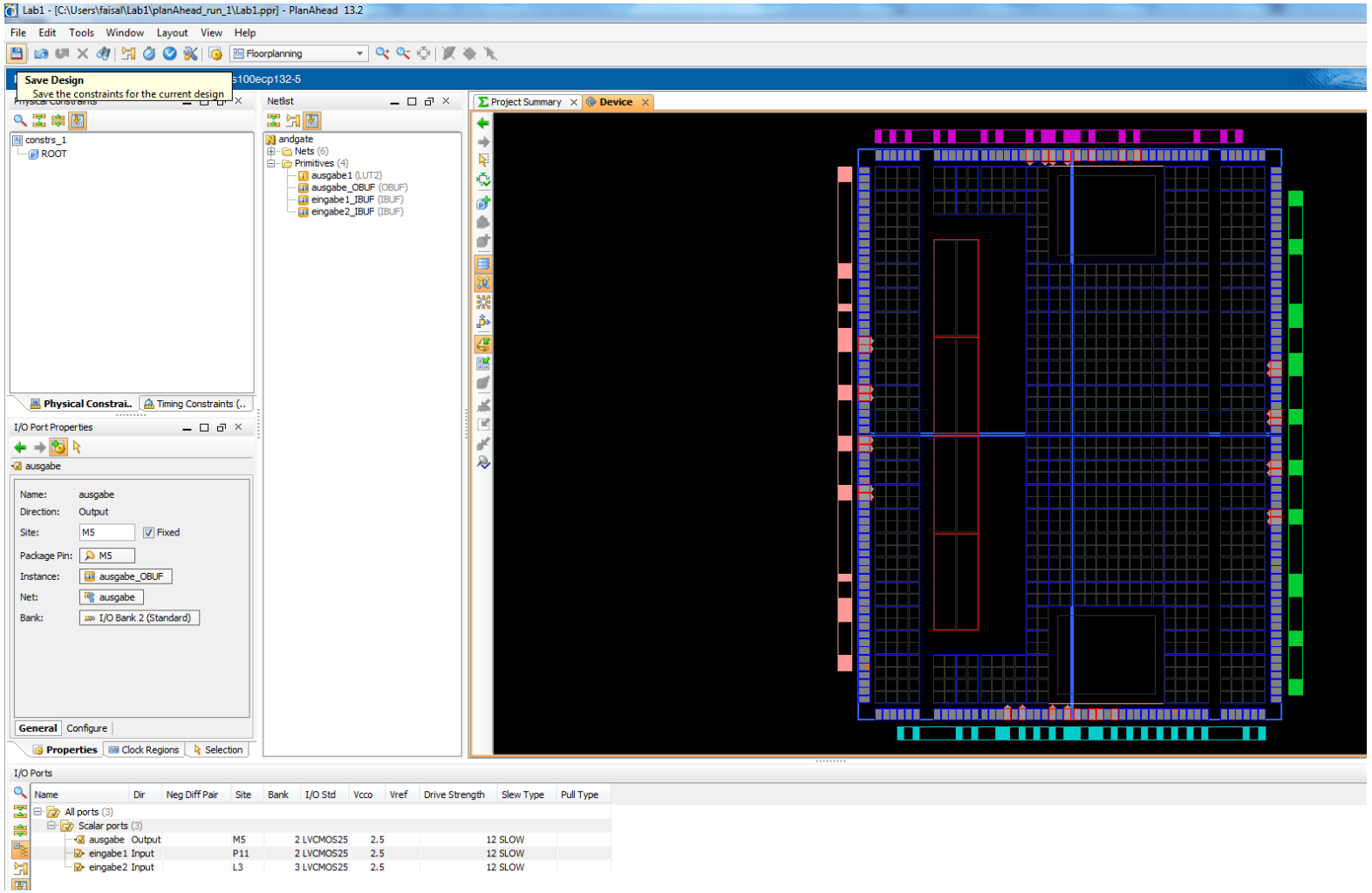
18. Click the I/O ports Tab



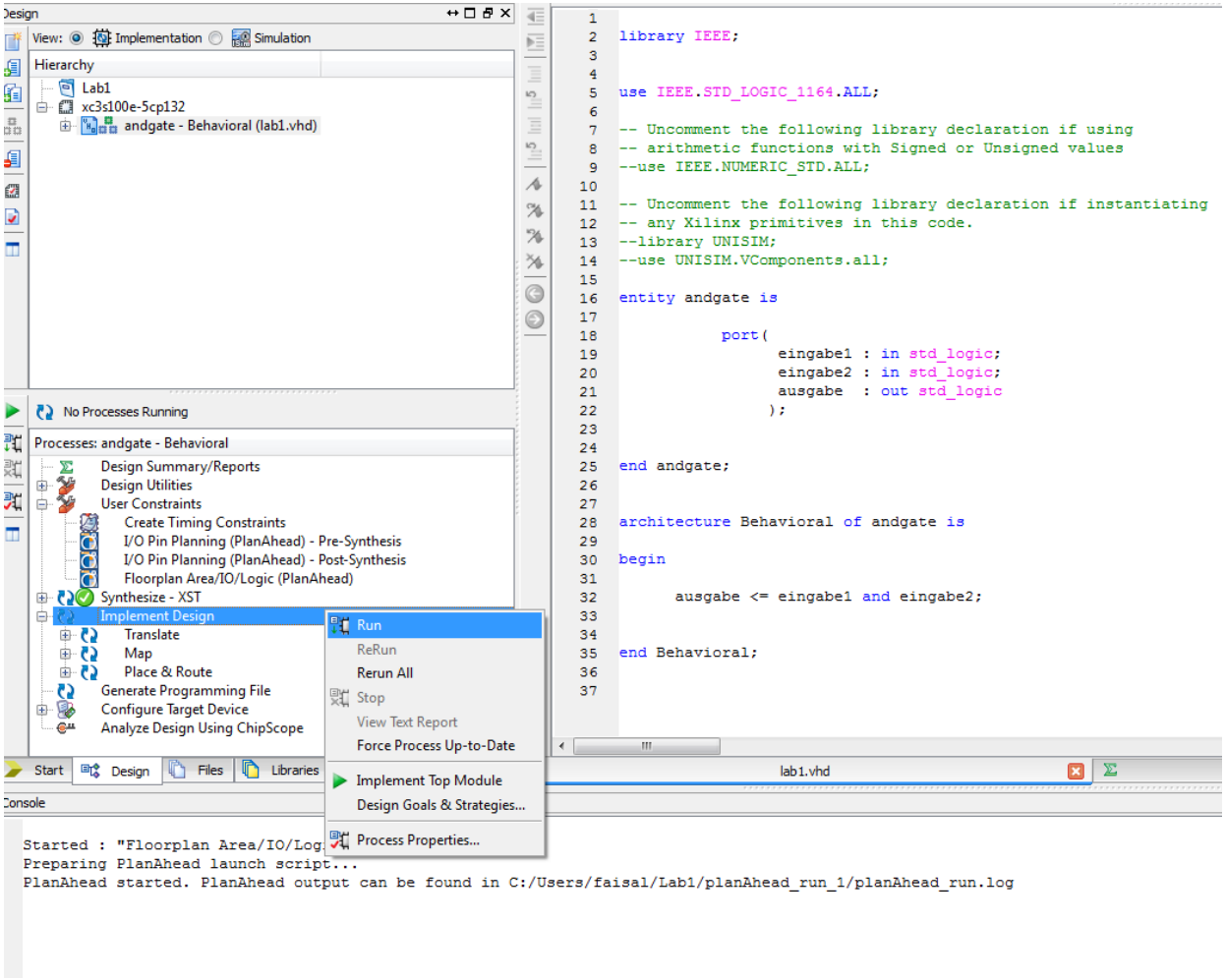
19. Select the Appropriate Pins for each of the Port Signal from the Site Column by double clicking it corresponding to the Signal.



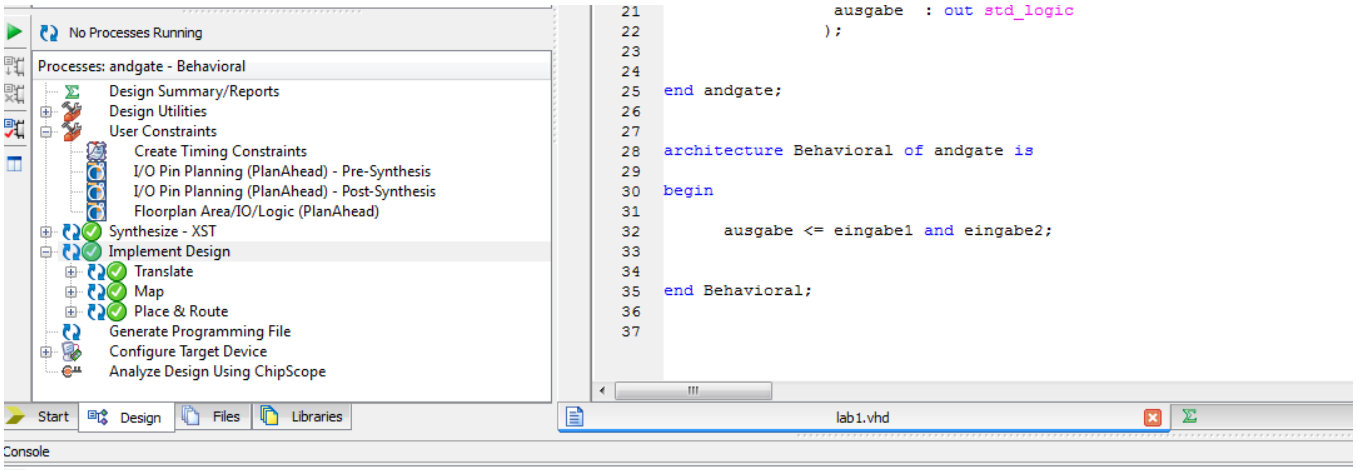
20a. When you are done with the pin Assignment , Click to save the changes and switch to the ISE design Software.



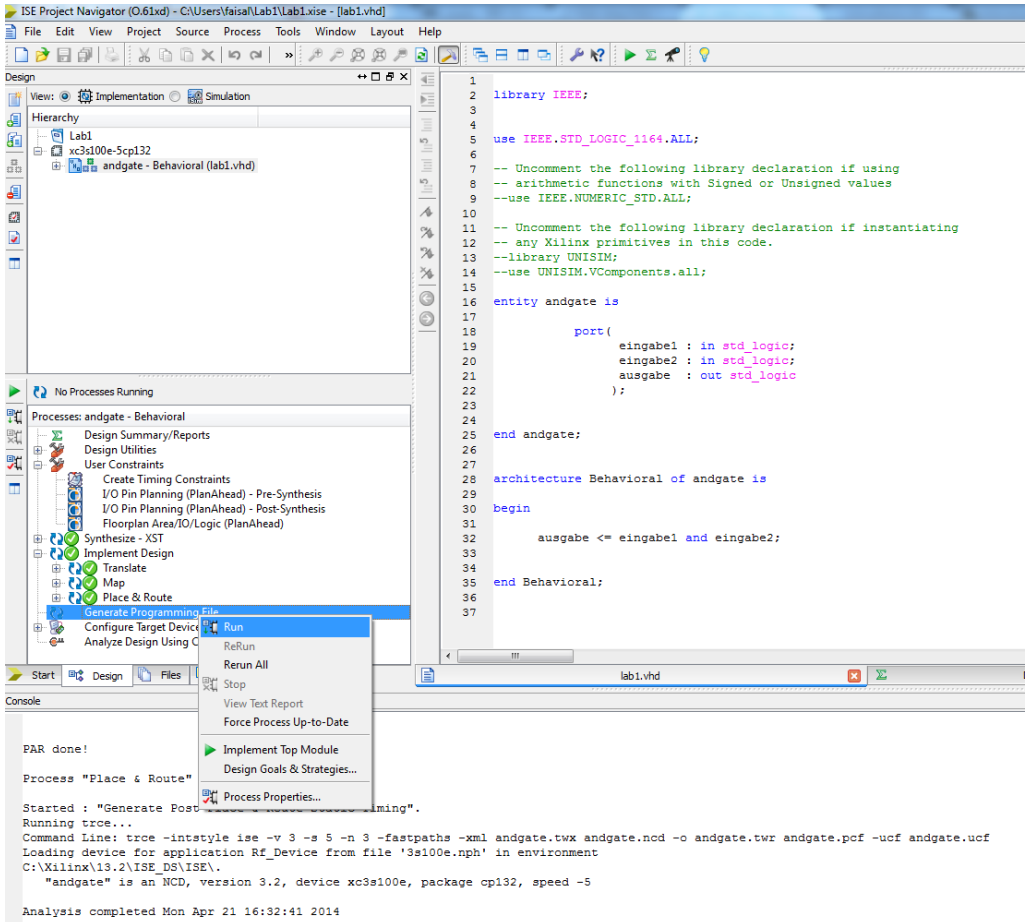
20b. Now right click on the Implement design and click run to Translate, Map , Place and Route



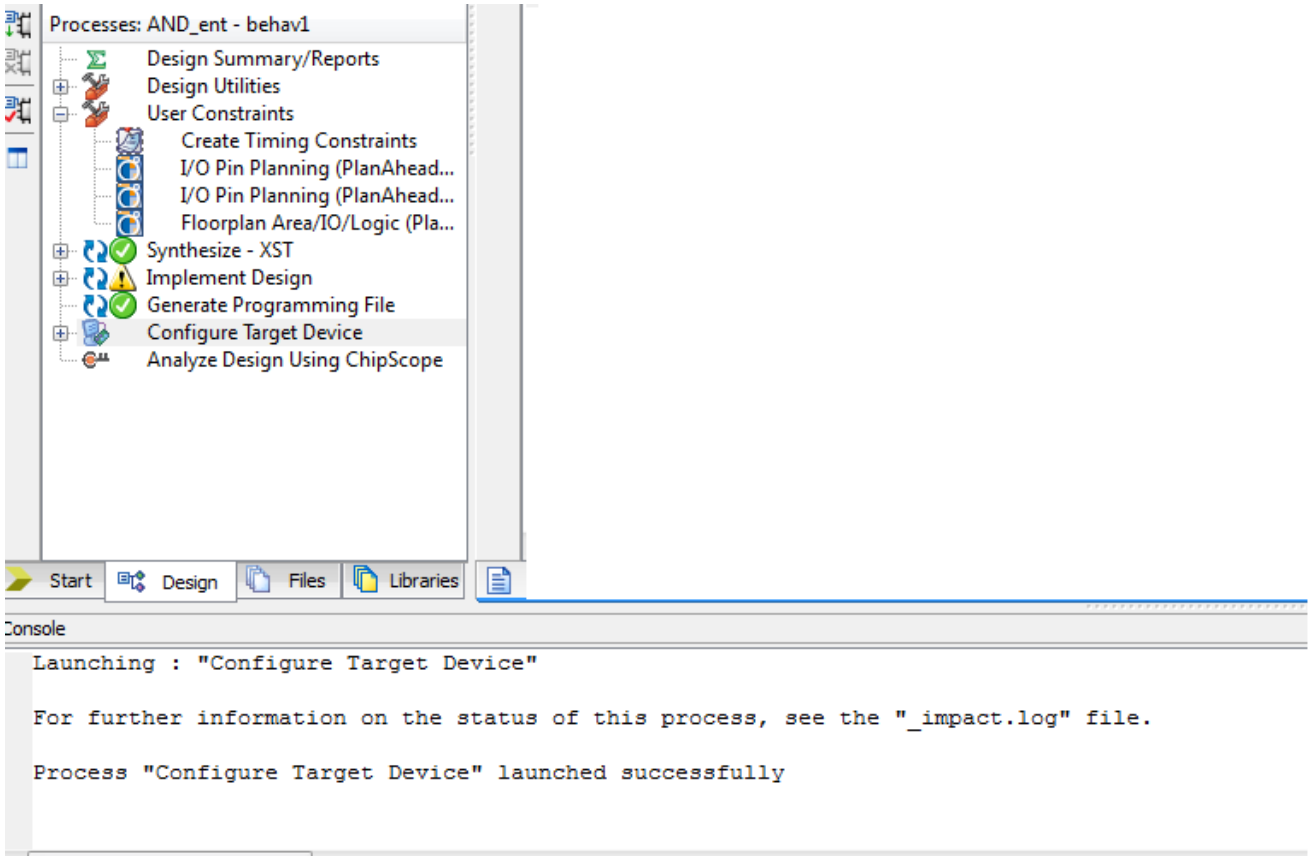
21. When the design is implemented successfully it will be shown up green as follows.



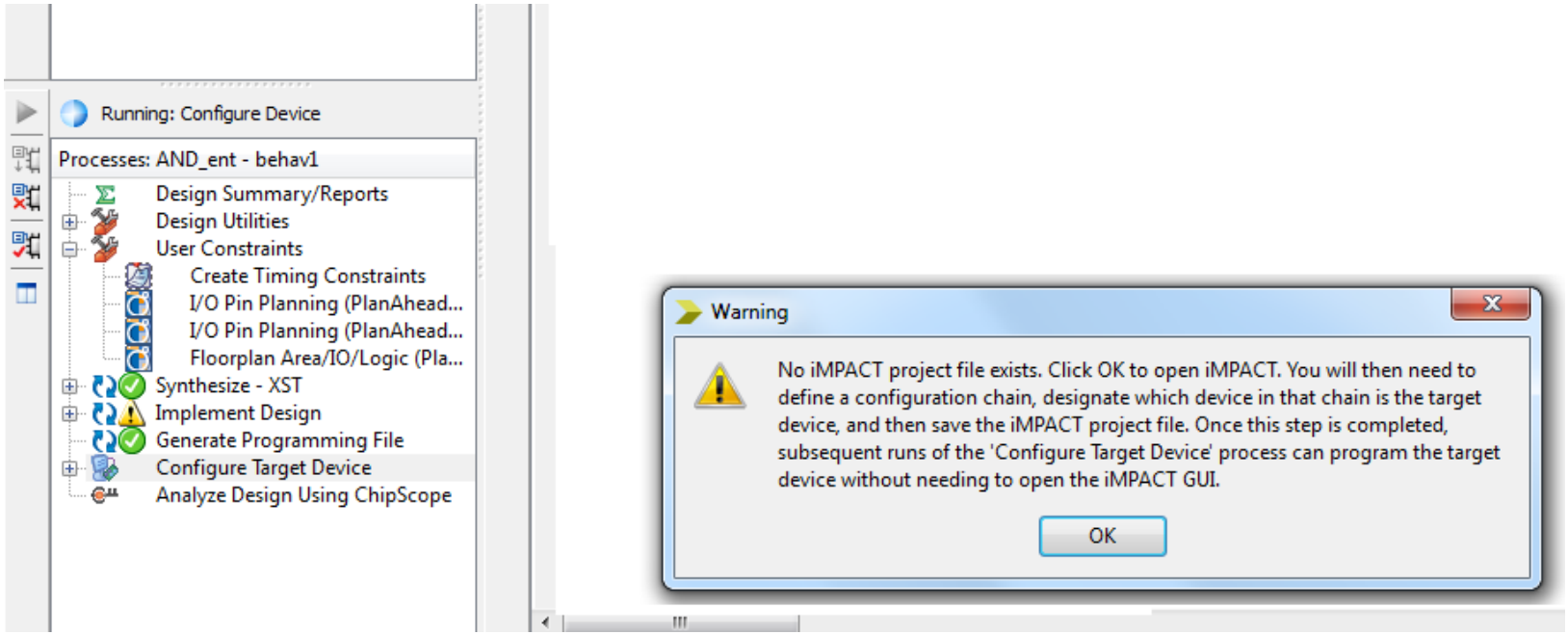
20. Now you have to Generate the programming File so right click on it and run.



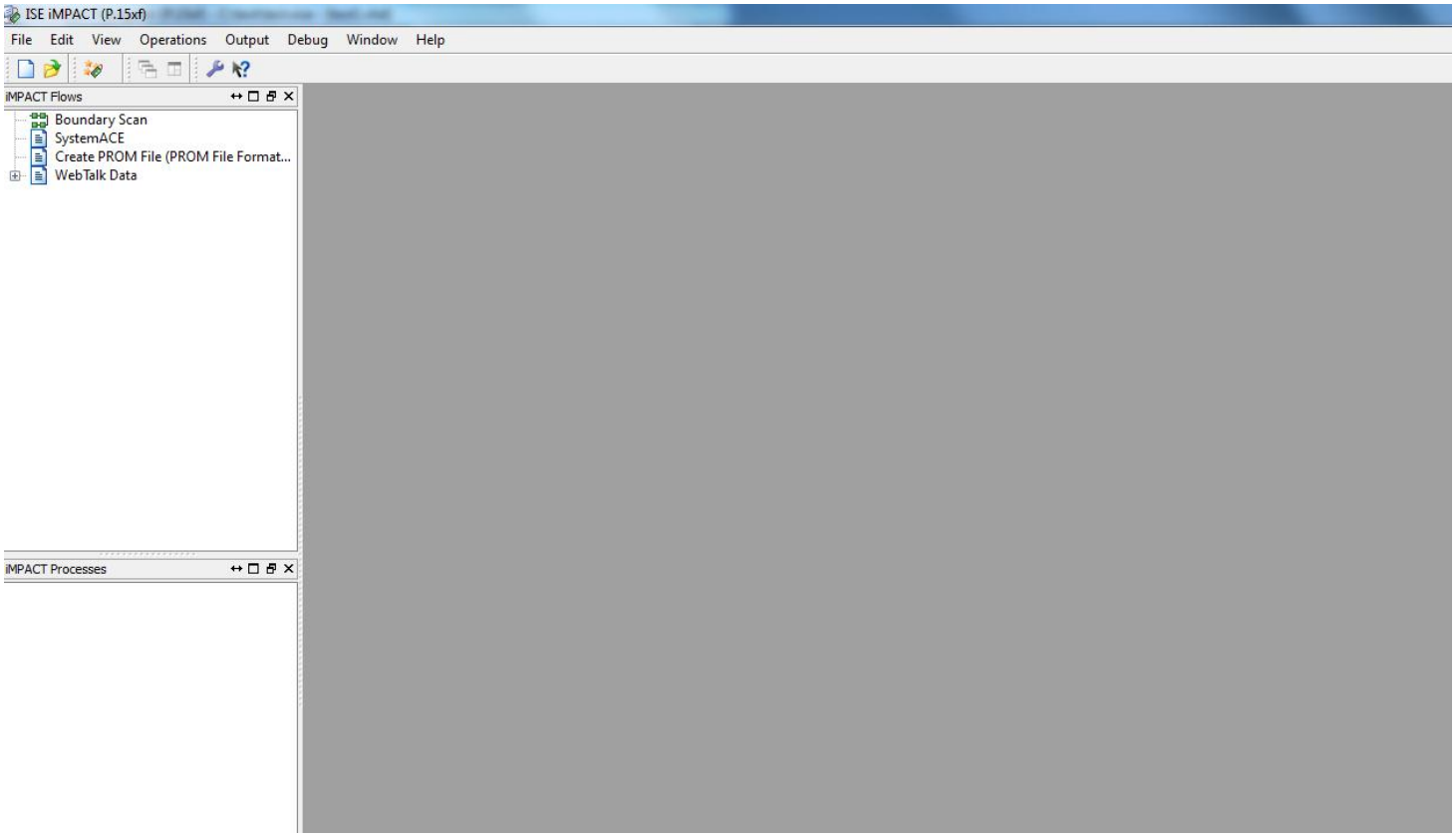
22. If this process is successfully completed then you will have a .bit file in your project directory which will downloaded on the FPGA kit.



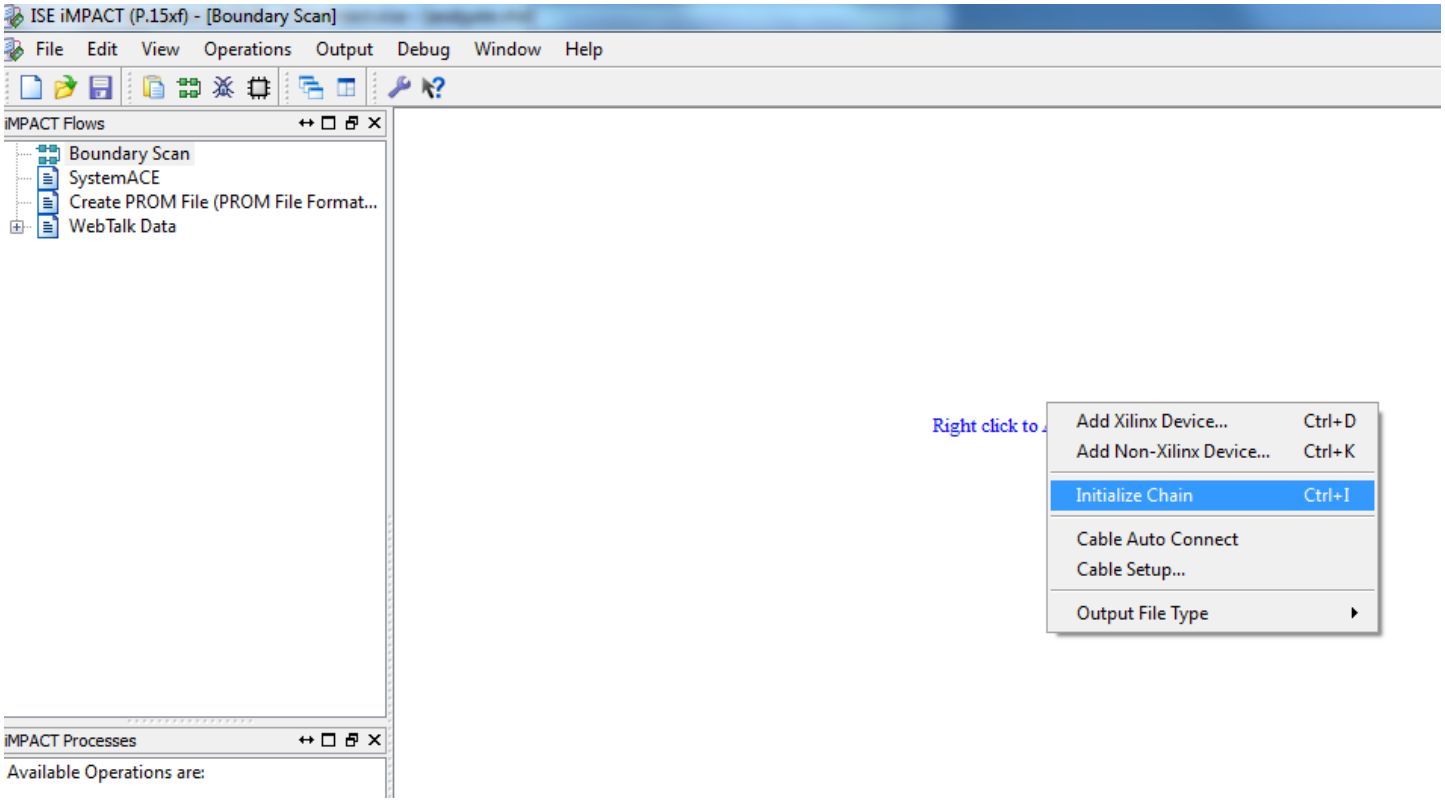
22. Now it’s the turn of the downloading procedure so double click on Configure Target Device.



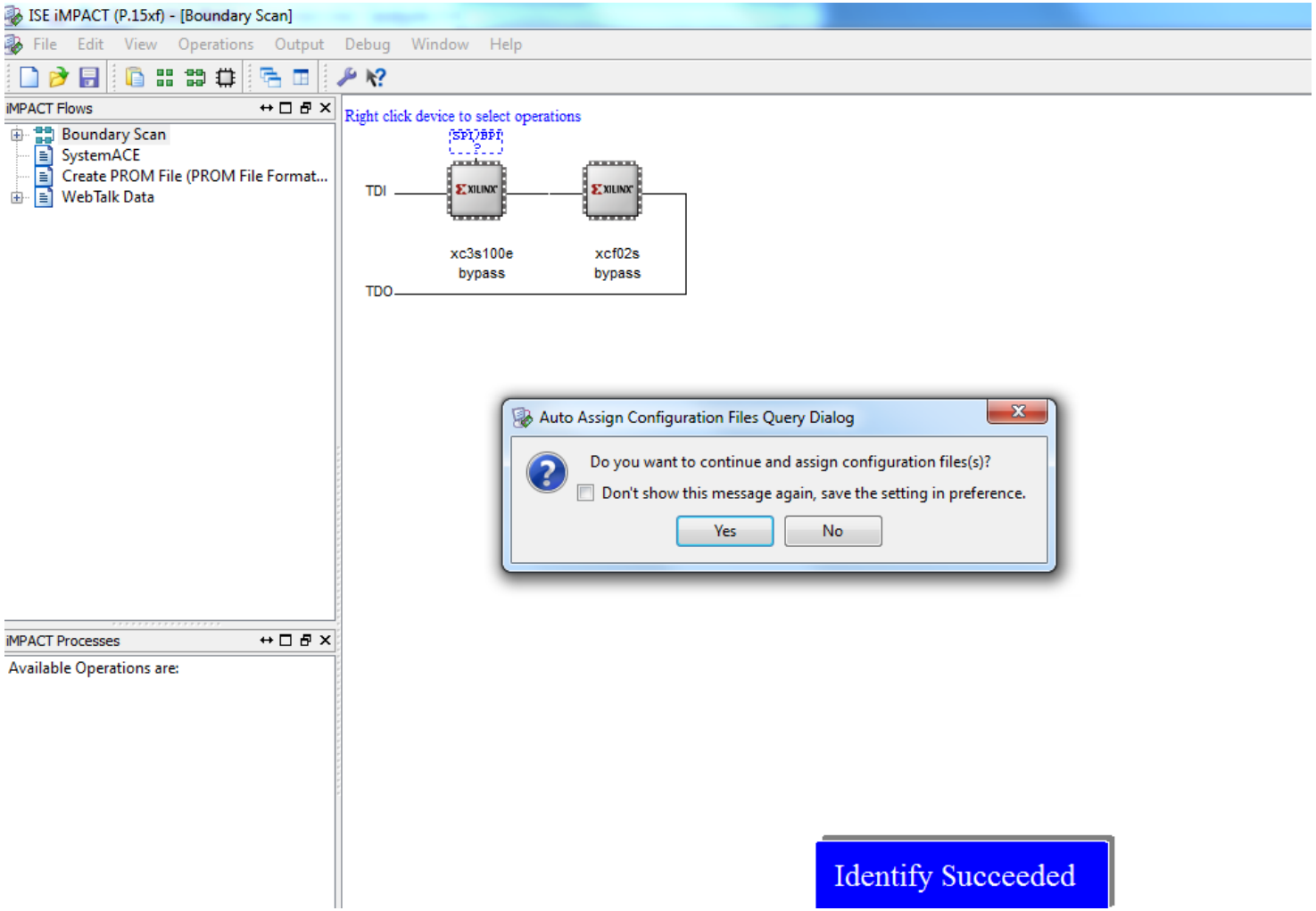
23. Double Click on Boundary Scan.



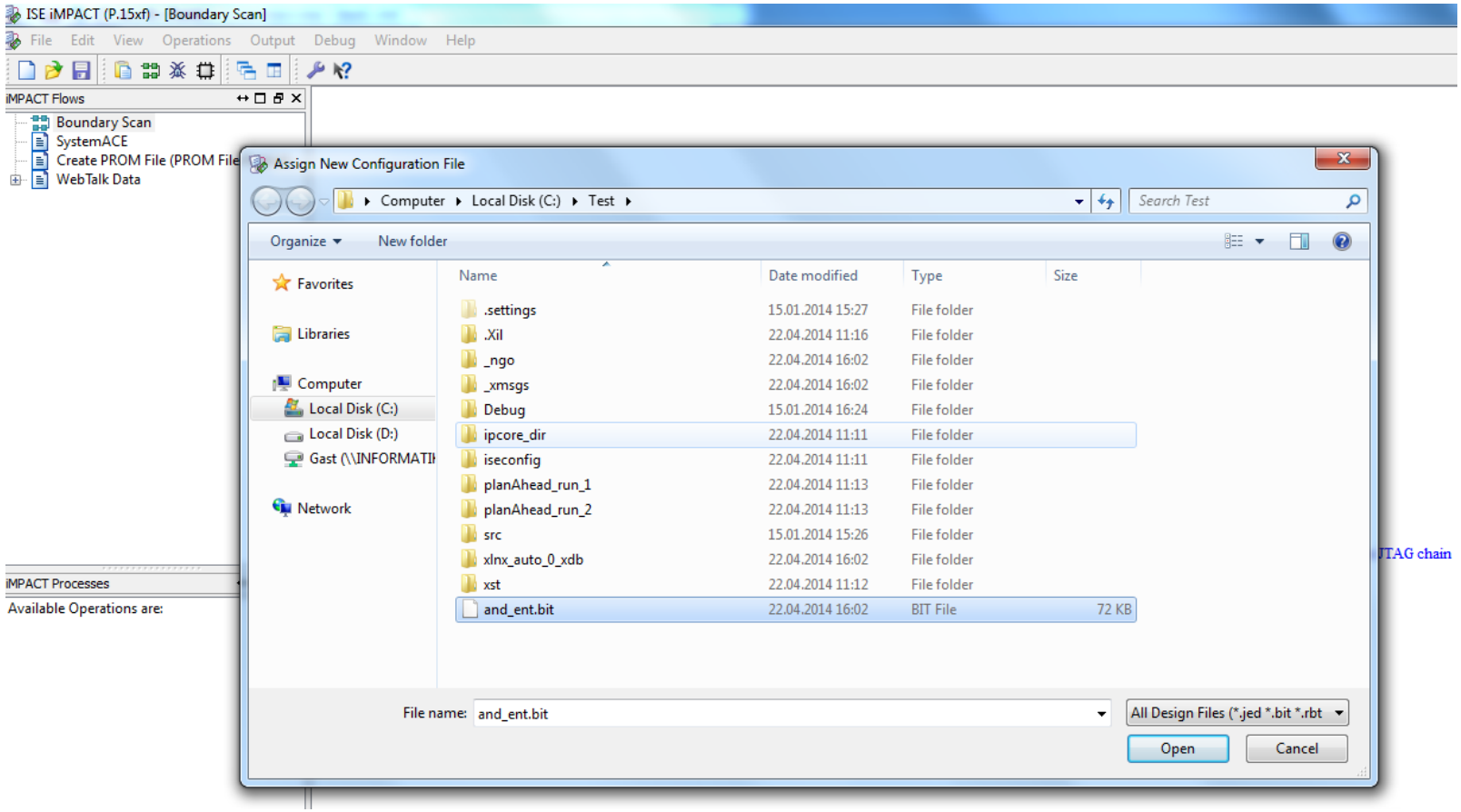
24a. Follow the instruction on the white window and select the Initialize Chain option.



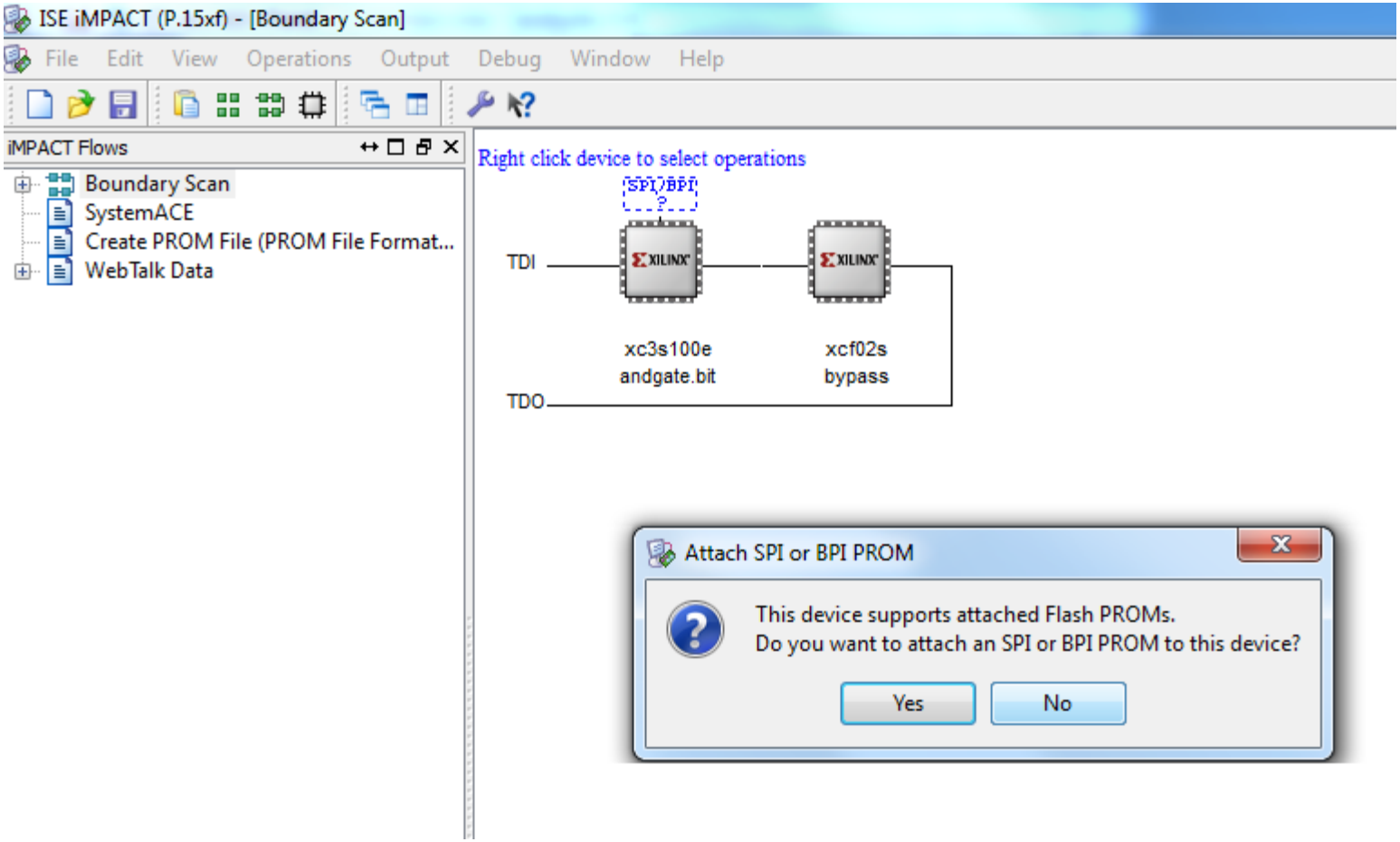
24b. If board is connected and Switched On then you will have the following output



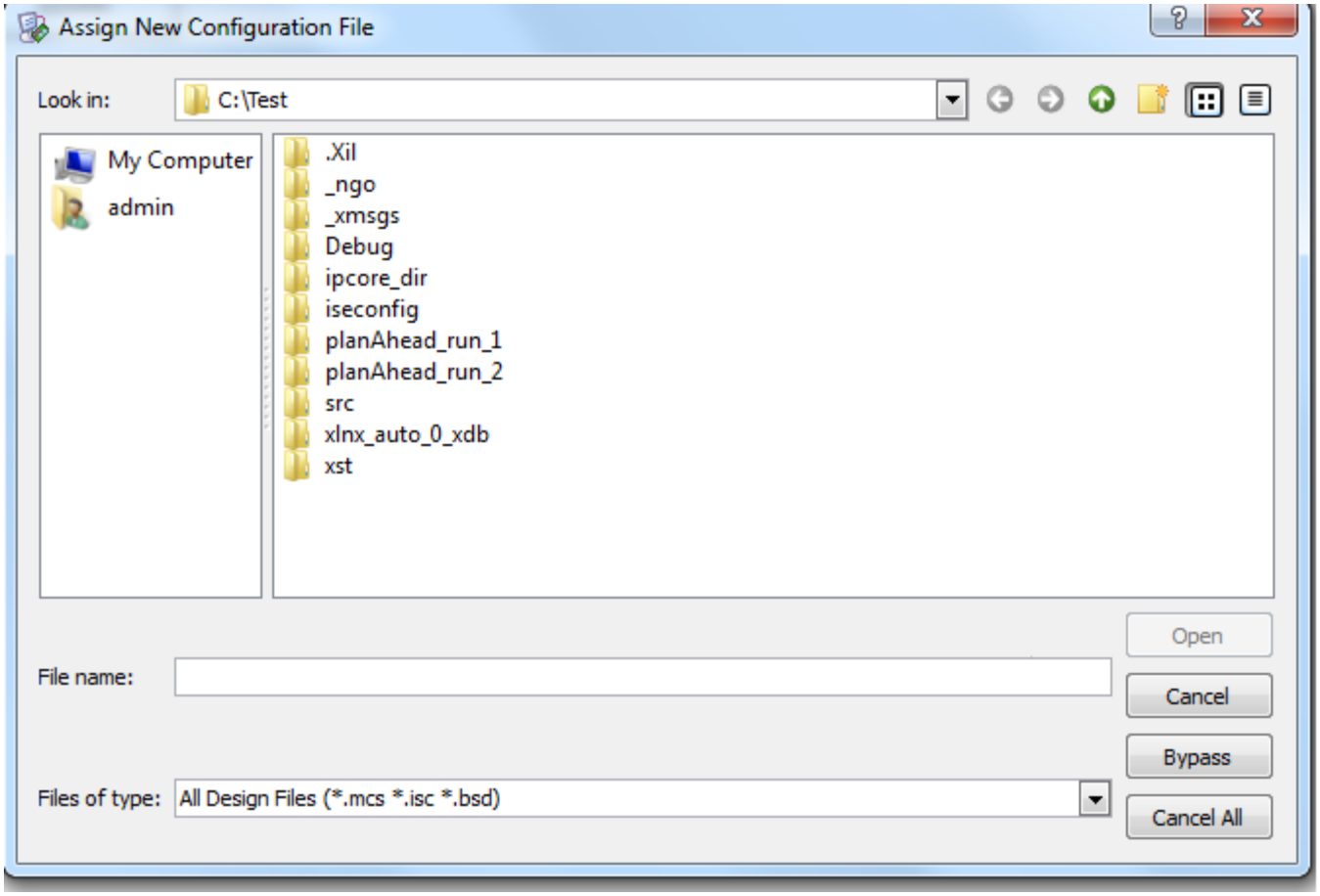
25. Now select the bit file and click open.



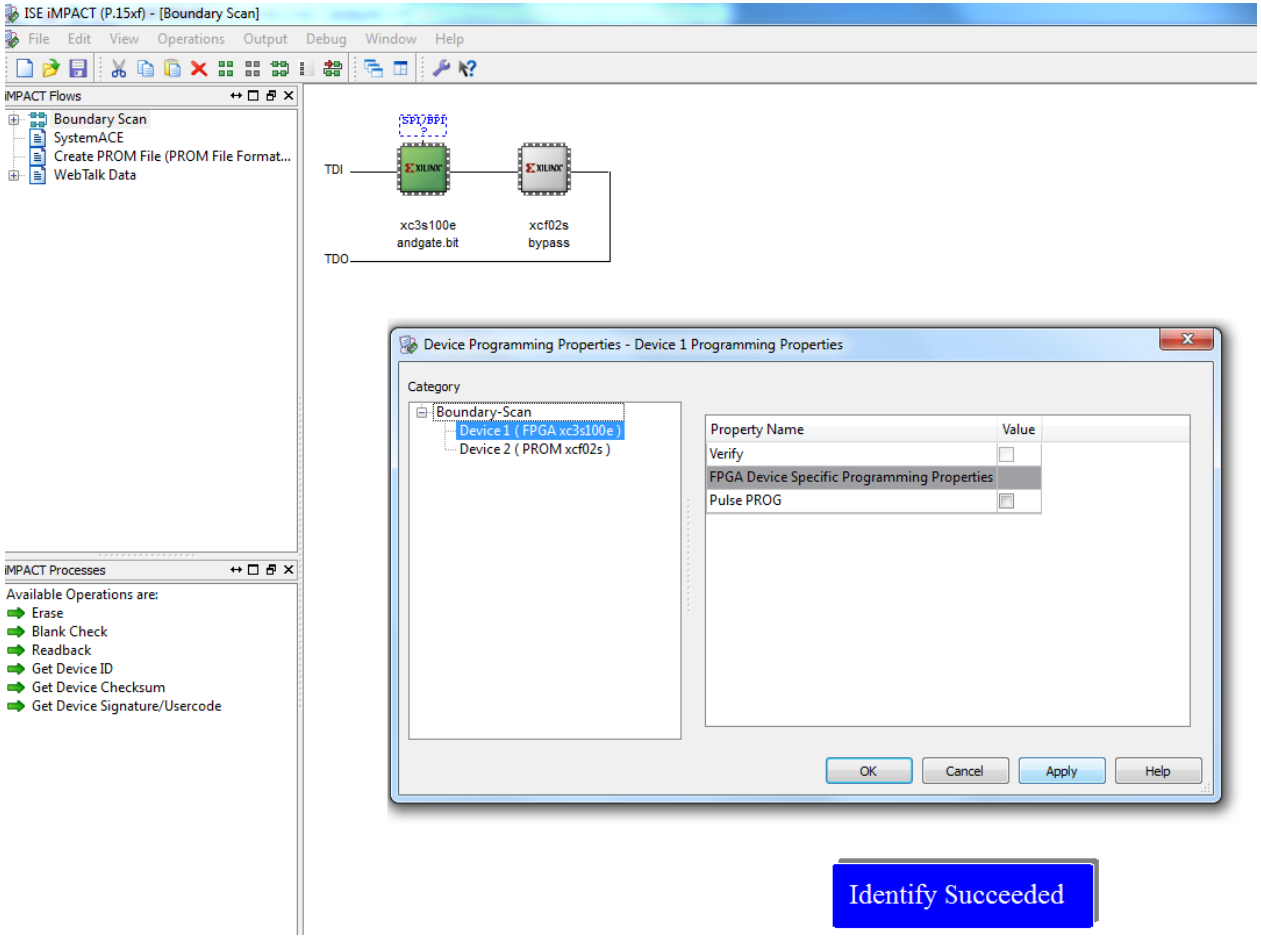
We are not using PROMS so click No.



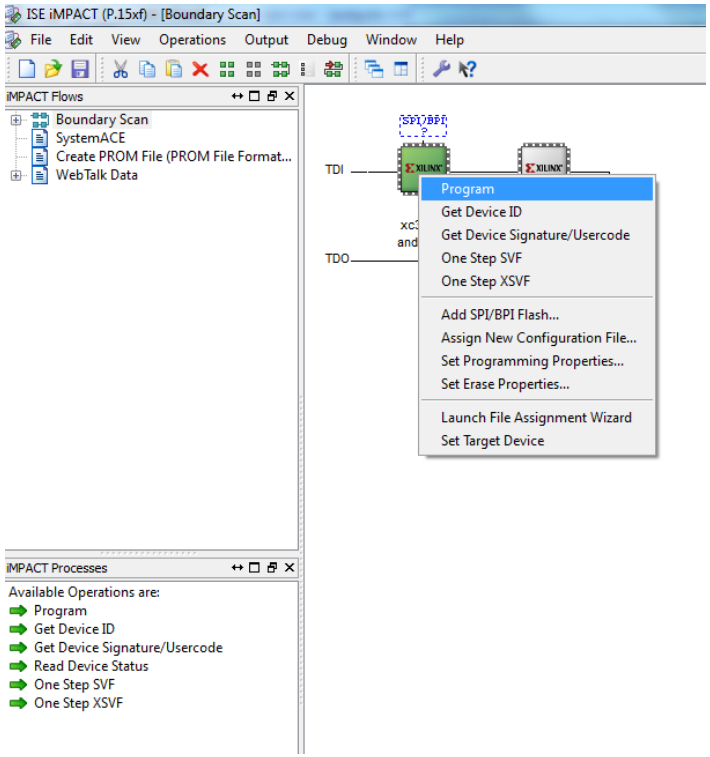
26a. Click Bypass



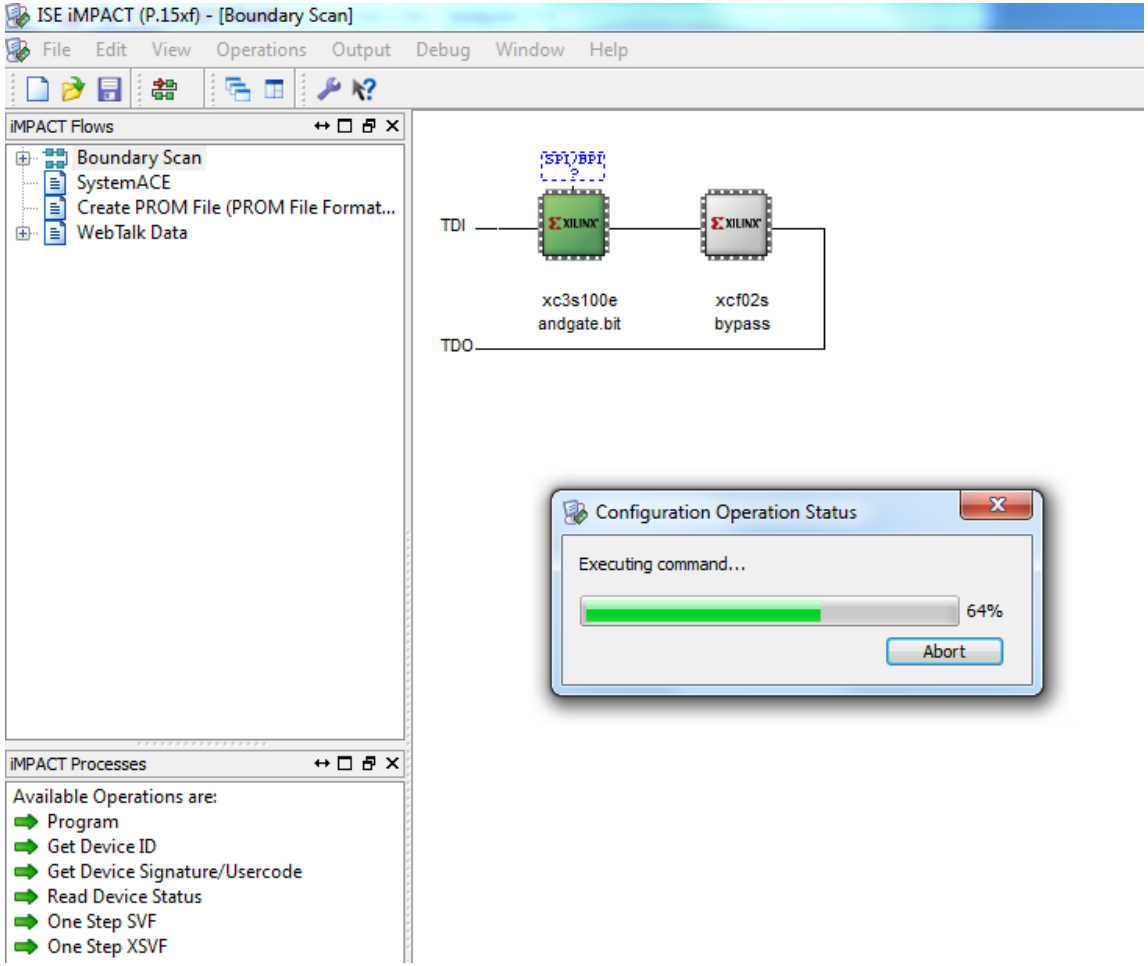
26b.Click Ok .


ISE iMPACT (P.15xf) - [Boundary Scan]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PROM File Format...
WebTalk Data
TDI
xc3s100e
andgate.bit
xcf02s
bypass
TDO
iMPACT Processes
Available Operations are:
Erase
Blank Check
Readback
Get Device ID
Get Device Checksum
Get Device Signature/Usercode
Device Programming Properties - Device 1 Programming Properties
Category
Boundary-Scan
Device 1 (FPGA xc3s100e)
Device 2 (PROM xcf02s)
Property Name
Value
Verify
FPGA Device Specific Programming Properties
Pulse PROG
OK
Cancel
Apply
Help
Identify Succeeded

27a. Right Click on the device, It will be turned Green, Now select Program.



27b.You will see a red light flashing and status bar.



28. If the program is successfully downloaded on the board then the output will appear as shown below.